



Universidad Carlos III de Madrid

Ingeniería Superior en Informática

PROYECTO FIN DE CARRERA

**ENTORNO DE AUTORÍA DE
MATERIALES DIDÁCTICOS
BASADOS EN IMS LEARNING
DESIGN**

Autor: Carlos Fernández Viso

Tutora: Carmen L. Padrón Nápoles

Ingeniería Informática Superior

Agradecimientos

Para comenzar, quiero agradecer a mi tutora Carmen Padrón todo su trabajo para que este proyecto haya podido llevarse a cabo, ya que de no ser por ella, hubiera sido totalmente imposible la consecución con éxito del mismo, ya que han surgido múltiples imprevistos que han dificultado su realización, pero ella siempre a puesto todo de su parte para que se llevara a buen puerto.

Quiero mostrar mi agradecimiento también a todo el personal docente con el que he coincidido en estos años en la Universidad, ya que gracias a ellos he adquirido una gran cantidad de conocimientos que me han permitido poder afrontar la realización de tareas del tipo de las que abarca este proyecto con garantías de poder superarlo.

En otro orden de cosas, quiero darle las gracias a todos mis amigos, tanto los de toda la vida que conozco de antes de iniciar mi formación universitaria, como a los que he conocido durante la misma, gente que me ha apoyado en todo lo que estaba en su mano y que ha hecho que disfrute la experiencia de la Universidad y la viva desde otro punto de vista.

Y por supuesto, quiero aprovechar para agradecer su apoyo a mi familia, especialmente a los más cercanos, a mi padre por enseñarme que con trabajo todo llega, a mi madre por estar siempre ahí para mí, en cualquier cosa que he necesitado, a mi hermana por darme consejos que siempre me han servido para ir hacia delante, y a todos ellos en general por todo lo que tuvieron que sacrificar y adaptar para que yo pudiera estudiar de la forma más tranquila y completa posible, estando ahí en todo momento.

Muchas Gracias a todos.

ÍNDICE

1. Introducción	8
1.1. Objetivos.....	12
1.2. Solución desarrollada	13
2. Estado del Arte	15
2.1. El concepto e-learning	15
2.2. Estandarización	18
2.3. Iniciativas de Estándares de e-learning	22
2.3.1. AICC, Aviation Industry CBT Committe.....	22
2.3.2. IEEE Learning Technologies Standards Committee (LTSC)	22
2.3.3. IMS Global Learning Consortium, Inc	22
2.3.4. ADL SCORM.....	32
2.4. Plataformas de enseñanza (LMS).....	34
3. Planteamiento del Problema	37
3.1. Herramientas de autoría.....	37
3.1.1. RELOAD	37
3.1.1.2. MD2: Método de desarrollo de materiales didácticos	45
3.2. Tecnologías utilizadas	47
3.2.1. XML	47
3.2.2. JDOM	49
3.2.3. JBoss	50
3.2.4. Struts	51
3.2.5. Coppercore y SLED.....	52
4. Análisis del sistema	55
4.1. Finalidad del sistema.....	55
4.2. Requisitos de usuario.....	57
4.3. Composición de Materiales Didácticos	65
5. Herramienta de composición de materiales didácticos	69
5.1. Arquitectura	69
5.2. Casos de uso.....	72
5.3. Diseño detallado de la aplicación.....	75
5.3.1. Diseño Gráfico de la Aplicación	75
5.3.2. Diseño técnico de la Aplicación.....	77
6. Manual de Configuración	92
6.1. Mensajes de la aplicación	92
6.1.2. Mensajes mostrados en la cabecera de la pantalla y en la barra situada debajo de la cabecera de la pantalla:	93
6.1.3. Mensajes mostrados en el pie de la pantalla:	94
6.1.4. Mensajes mostrados en la pantalla de introducción de descripciones de actividades:	94
6.1.5. Mensajes mostrados en la pantalla que indica el éxito en el proceso de composición:.....	95
6.1.6. Mensajes que se muestran en la pantalla de error en el proceso de composición:.....	96

6.2. Estilo visual de la aplicación	97
6.3. Repositorio de la aplicación.....	97
6.4. Recursos obligatorios en las unidades.....	98
6.5. Despliegue de la aplicación WEB.....	98
7. Manual de Usuario	100
7.1. Visión general	100
7.2. Objetivos de la aplicación.....	101
7.3. Guía de funcionalidades	108
7.3.1. Selección de Recursos	108
7.3.2. Introducción de descripciones de actividades	122
7.3.3. Finalización de la composición.....	125
8. Gestión del Proyecto	127
8.1. Análisis de Costes	131
8.1.1. Personal.....	131
8.1.2. Material e Infraestructuras	132
8.1.3. Otros	133
8.1.4. Resumen de Costes	134
9. Conclusiones	135
9.1. Evaluación	135
9.2. Conclusiones del Proyecto.....	142
9.3. Ampliaciones Futuras.....	143
10. Bibliografía	144

Tabla de Ilustraciones

Figura 1: Esquema básico de funcionamiento de un sistema de e-learning.....	17
Figura 2: Esquema de aplicaciones de soporte al e-learning	18
Figura 3: Esquema de actores de un sistema de e-learning.....	20
Figura 4: Esquema de la estructura de un paquete de contenidos del estándar IMS Content Package	26
Figura 5: Estructura de un fichero de manifiesto basada en IMS Learning Design según MD2 (parte 1).	29
Figura 6: Estructura de un fichero de manifiesto de IMS Learning Design según MD2 (parte 2).....	30
Figura 7: División en libros de la actual versión de SCORM.....	32
Figura 8: Herramienta RELOAD (parte 1)	37
Figura 9: Herramienta RELOAD (parte 2)	38
Figura 10: Herramienta RELOAD (parte 3)	39
Figura 11: Herramienta RELOAD (parte 4)	40
Figura 12: Herramienta RELOAD (parte 5)	41
Figura 13: Herramienta RELOAD (parte 6)	42
Figura 14: Herramienta RELOAD (parte 7)	43
Figura 15: Estructura de la interacción entre SLED y Coppercore	53
Figura 16: Estructura del Modelo-Vista-Controlador.....	69
Figura 17: Representación del patrón MVC y su interacción con el usuario.....	71
Figura 18: Caso de Uso 1: Visualizar Materiales Disponibles.....	72
Figura 19: Caso de uso 2: Examinar Materiales Didácticos	73
Figura 20: Caso de uso 3: Seleccionar Materiales Didácticos para Composición y Realizar Composición.....	73
Figura 21: Caso de uso 4: Publicar Paquetes de Contenidos creados.....	74
Figura 22: Caso de uso 5: Visualizar Paquete de Contenidos creado.....	74
Figura 23: Diseño de la interfaz de usuario de la aplicación	76
Figura 24: Módulo de Unidades de Aprendizaje.....	78
Figura 25: Módulo de Fábricas de Unidades.....	80
Figura 26: Módulo de Identificadores de elementos	82
Figura 27: Módulo de Utilidades, Tabla de Relaciones.....	84
Figura 28: Módulo de Utilidades, Clases alternativas	85
Figura 29: Módulo de Utilidades, Entrada	86
Figura 30: Módulo de Utilidades. Barra de Progreso	87
Figura 31: Módulo de Utilidades, Excepción.....	88
Figura 32: Módulo de Componentes Web, Servlet Composición.....	88
Figura 33: Módulo de Componentes Web, Servlet VerRecurso	89
Figura 34: Módulo de Componentes Web, Servlet ComposicionDescripciones	89
Figura 35: Módulo de Componentes Web, Servlet Progreso.....	90
Figura 36: Módulo de Componentes Web, Servlet AccionSiguiente.....	90
Figura 37: Módulo de Componentes Web, Filtro Composición	91
Figura 38: Mensajes de la pantalla de selección de recursos	92
Figura 39: Mensajes mostrados en la cabecera de la pantalla y en la barra situada debajo de la cabecera de la pantalla.....	93

Figura 40: Mensajes mostrados en el pie de la pantalla.....	94
Figura 41: Mensajes mostrados en la pantalla de introducción de descripciones de actividades	94
Figura 42: Mensajes mostrados en la pantalla que indica el éxito en el proceso de composición.....	95
Figura 43: Mensajes que se muestran en la pantalla de error en el proceso de composición.....	96
Figura 44: Ejemplo del interior de una unidad de aprendizaje.....	102
Figura 45: Ejemplo de archivo de manifiesto – parte 1 - Actividades	103
Figura 46: Ejemplo de archivo de manifiesto – parte 2 - Actividades	104
Figura 47: Ejemplo de archivo de manifiesto – parte 3 – Actividades.....	105
Figura 48: Ejemplo de archivo de manifiesto – parte 4 – Environments.....	106
Figura 49: Ejemplo de archivo de manifiesto – parte 5 – Play	106
Figura 50: Ejemplo de archivo de manifiesto – parte 6 – Recursos	107
Figura 51: Pantalla de Selección de Recursos (1)	108
Figura 52: Pantalla de Selección de Recursos (2)	109
Figura 53: Pantalla de Selección de Recursos (3)	110
Figura 54: Pantalla de Selección de Recursos (4)	111
Figura 55: Pantalla de Selección de Recursos (5)	112
Figura 56: Pantalla de Selección de Recursos (6)	113
Figura 57: Pantalla de Selección de Recursos (Visualización 1)	114
Figura 58: Pantalla de Selección de Recursos (Visualización 2)	115
Figura 59: Pantalla de Selección de Recursos (Descarga 1)	116
Figura 60: Pantalla de Selección de Recursos (Descarga 2)	117
Figura 61: Pantalla de Selección de Recursos (Descarga 3)	118
Figura 62: Pantalla de Selección de Recursos (Error 1)	119
Figura 63: Pantalla de Selección de Recursos (Error 2)	120
Figura 64: Pantalla de Composición (Barra de Progreso).....	121
Figura 65: Introducción de descripciones de actividades (1).....	122
Figura 66: Introducción de descripciones de actividades (2).....	123
Figura 67: Introducción de descripciones de actividades (3).....	123
Figura 68: Introducción de descripciones de actividades (4).....	124
Figura 69: Pantalla de Finalización de la composición (1)	125
Figura 70: Pantalla de Finalización de la composición (2)	125
Figura 71: Pantalla de Finalización de la composición (3)	126
Figura 72: Finalización de la composición (4).....	126
Figura 73: Planificación de tareas del Proyecto	128
Figura 74: Diagrama Gantt del Proyecto	130
Figura 75: Tiempos del personal implicado en el Proyecto	132
Figura 76: Resumen de días y costes totales del personal del Proyecto	132
Figura 77: Costes de materiales e infraestructuras	133
Figura 78: Otros gastos	133
Figura 79: Resumen de costes del Proyecto.....	134
Figura 80: Detalles de las pruebas realizadas para la evaluación del método propuesto por MD2.	136
Figura 81: Resultados asociados al proceso de Composición-Diseño XML	137

Figura 82: Resultados asociados a la disposición de contenidos, estrategia y estructura	138
Figura 83: Resultados asociados al proceso de integración	139
Figura 84: Resultados asociados a la ayuda a la visualización	139
Figura 85: Resultados del proceso de evaluación.....	140
Figura 86: Resultados asociados al proceso de desarrollo con las herramientas	141
Figura 87: Resultados del estudio del tiempo empleado en el desarrollo.....	141

1. Introducción

Los contenidos educativos en formato digital son una parte fundamental en los nuevos modelos de enseñanza que se imponen, los catalogados como e-learning. El e-learning se define de diversas formas, debido a la variedad de los usuarios del mismo, que hacen uso de él cada uno en su ámbito y para sus necesidades concretas en el ámbito que nos ocupa, la enseñanza digital.

Si lo analizamos desde una perspectiva de su concepción y desarrollo como herramienta formativa, se diferencian dos puntos de vista fundamentales, una pedagógica y otra tecnológica. La pedagógica se refiere a que estos sistemas no deben ser meros contenedores de información digital, sino que deben estar estructurados de forma que sigan unos modelos y patrones, que deben ser definidos de manera que se pueda afrontar el reto de estos nuevos contextos. La tecnológica en cuanto a que se está tratando con software, aplicaciones WEB principalmente que es lo que otorga a estos sistemas el sobrenombre de plataformas de formación, dedicadas a difundir conocimientos a sus usuarios.

Si atendemos a su uso, podríamos hablar en función del uso que le van a dar sus usuarios finales, independientemente de la formación y la madurez de los mismos, que ven en el sistema e-learning como el medio para alcanzar sus objetivos en su formación. Pero también es posible observar el punto de vista de una organización, en la que con unos objetivos definidos y el alcance de la formación basada en estos sistemas, se distingue una visión académica y una visión empresarial.

El origen de la palabra e-learning hace referencia al aprendizaje electrónico, que como tal, puede abarcar cualquier actividad educativa que haga uso de los medios electrónicos para completar todo el proceso de formación.

Desde una perspectiva basada en la experiencia en el desarrollo y utilización de las plataformas e-learning, el profesor Francisco José García Peñalvo, de la Universidad de Salamanca, explica el e-learning como *la capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada discente, además de garantizar ambientes de aprendizaje colaborativo mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias.* [1]

Otras definiciones provocan una aplicación del concepto de e-learning a casi cualquier proceso relacionado con educación y tecnologías, como la definición de American Society of Training and Development que define el e-learning como *el término que cubre un amplio grupo de aplicaciones y procesos, tales como aprendizaje basado en Web, aprendizaje basado en ordenadores, aulas virtuales y colaboración digital. Incluye entregas de contenido vía Internet, intranet/extranet,*

audio y video grabaciones, transmisiones satelitales, TV interactiva, CD-ROM y más. [2]

Otras definiciones evitan dar una visión tan amplia del término e-learning, y realizan una definición más reducida al ámbito de Internet, encontrando definiciones como: “el uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento. Está basado en tres criterios fundamentales: [3]

1. El e-learning trabaja en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información.

2. Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet.

3. Se enfoca en la visión más amplia del aprendizaje que van más allá de los paradigmas tradicionales de capacitación. Esta definición es dada por Rosenberg en el año 2001.

Todas estas definiciones, así como otras muchas a las que se puede acceder a través de la bibliografía especializada, tienen en común la referencia explícita o implícita a lo que se ha denominado *el triángulo del e-learning*[4] formado por la tecnología(plataformas, campus virtuales...), los contenidos(la calidad y estructuración de los mismos es condición imprescindible para el éxito de estas iniciativas de e-formación) y los servicios(siendo este el elemento más variado puesto que en el se engloba la acción de los profesores, elementos de gestión, elementos de comunicación, elementos de evaluación...). Si se varía el peso de estos elementos se obtienen diferentes modelos de e-formación, un proceso similar al que sigue un profesor variando sus variables y recursos para obtener diferentes políticas de docencia personal adecuadas a su objetivo concreto.

A la hora de llevar a cabo un programa de formación basado en e-learning, se hace uso de plataformas o software que permita la comunicación entre los implicados en el proceso, es decir, profesores, alumnos y contenidos. Podemos encontrar dos tipos de plataformas: las que se utilizan para impartir y realizar un seguimiento administrativo de cursos online o **LMS(Learning Management Systems)**, y las que son utilizadas con el objetivo de gestionar los contenidos digitales o **LCMS(Learning Content Management Systems)**.

Es necesario, para poder llevar a buen puerto la tarea de implantar esta nueva forma de crear contenidos, dada la heterogeneidad de plataformas educativas y de los sistemas de enseñanza en línea(los LMS), la existencia de estándares y recomendaciones aceptados por un grupo lo suficientemente representativo que permiten la interoperabilidad y reutilización de los OA, entre diferentes sistemas que coexisten.

Estándares

La principal función de los estándares en el ámbito del e-learning es, como no podía ser de otra manera, facilitar la interoperabilidad y la reutilización de los contenidos en el tiempo, es decir, posibilitar el que se puedan realizar intercambio de contenidos entre diferentes plataformas y sistemas. El término estándar tiende a provocar que en muchos casos se vean como un limitador de la creatividad personal de los autores, en este caso los educadores, que los ven como un aserie de normas de obligado cumplimiento que no les permite tener una total libertad en la creación de nuevos cursos y en su planificación propia de la tarea de elaboración de nuevos materiales. Otra circunstancia es asimilar que su uso es en la educación a distancia en exclusiva, y que no son útiles para otra forma de planteamientos educativos. Esto no es así, puesto que el seguir unas pautas en la creación de los materiales puede facilitar su uso posterior de forma que se simplifique el trabajo de los docentes, ya sea en la educación personal o a distancia

Las principales ventajas de la utilización de los estándares del e-learning para todas las partes implicadas en el proceso de aprendizaje son las que se enumeran a continuación:

- Atendiendo al punto de vista de los clientes o consumidores, tanto institucionales como individuales, los estándares evitan quedarse atrapados por las tecnologías propietarias. A su vez, se reducen costes al sustituir los desarrollos propios por otros de tipo “plug and play”, posibilitando de esta forma que, por ejemplo, una empresa pueda cambiar de LMS sin tener que empezar de cero todo el proceso perdiendo toda la información previamente acumulada.

- Atendiendo a los productores educativos, los estándares permiten que el formato de producción pueda ser utilizado en cualquier plataforma de distribución y que sea único. Es más, un mercado más amplio para los contenidos educativos permite a los creadores invertir en producción de contenidos, aumentando tanto la oferta como la calidad de estos, incluso en áreas con un nivel de especialización alto. La existencia de estándares facilita su labor, al permitir tener acceso a almacenes de contenidos reutilizables, y les permite la creación de contenidos modulares de más fácil y sencilla actualización y mantenimiento.

- Atendiendo a los vendedores de aplicaciones, la existencia de los métodos estandarizados de comunicación entre sistemas facilita la tarea de integración de diferentes productos, lo que provoca de manera directa una reducción de los costes de desarrollo e incrementa el mercado potencial para las aplicaciones.

- Atendiendo al punto de vista de los alumnos, los estándares provocan una mayor posibilidad de elección del producto educativo. Además, implica que los resultados del aprendizaje (créditos o certificados) tengan mayor portabilidad.

Existen otros trabajos en los que se destacan otras ventajas y propiedades beneficiosas del uso y aplicación de estándares (Marie 2003) [5]. Las ventajas se describen a continuación:

- **Interoperabilidad.** Que se pueda intercambiar y mezclar contenidos de múltiples fuentes y puedan ser usados en distintos sistemas. Que sistemas diferentes pueden comunicarse, intercambiar información e interactuar de forma transparente.

- **Gestionabilidad.** Que el sistema pueda obtener y trazar la información adecuada sobre el usuario y el contenido.

- **Durabilidad.** Que los consumidores no queden atrapados en una tecnología propietaria de una determinada empresa. Que no haya que hacer una inversión significativa para lograr la interoperabilidad o la reutilización.

- **Reusabilidad.** Que el contenido pueda ser agrupado, desagrupando reutilizando de forma sencilla y rápida. Que los objetos de contenido puedan ensamblarse y utilizarse en un contexto diferente de aquel para el que fueron inicialmente diseñados.

- **Accesibilidad.** Que un usuario pueda acceder al contenido adecuado en el momento que sea necesario y en el dispositivo diseñado para tal fin.

- **Escalabilidad.** Que las tecnologías sean configurables de manera que se permita aumentar la funcionalidad para dar servicio a más usuarios, respondiendo a las necesidades de la institución concreta y permitiendo que esto no provoque un desembolso desorbitado.

En puntos posteriores serán detallados los principales estándares existentes.

1.1. Objetivos

El objetivo del proyecto es realizar un estudio sobre los estándares actuales de e-learning, desde el punto de vista de la observación de la tecnología, plataformas y pedagogía que interviene en el mismo, entre otros factores. Por lo tanto, es necesario conocer la panorámica de la estandarización y profundizar en una especificación IMS LD como un paso previo, y plasmar los conocimientos adquiridos en una aplicación WEB que permita la composición, edición y visualización de contenidos educativos, bajo ese estándar.

La principal función de la aplicación será el poder componer materiales a partir de contenidos previamente almacenados en un repositorio.

La búsqueda se realiza en base a unos descriptores de objetivos de aprendizaje que contiene los conceptos a ser tratados. Dichos conceptos están relacionados con el tema a tratar.

La composición de los materiales didácticos se realizará a partir de la selección por parte del usuario de un subconjunto de los materiales recuperado de la búsqueda, que serán insertados siguiendo las plantillas definidas para cada uno de los tres tipos posibles de unidades de aprendizaje, Unidad de Simple Colaboración, Unidad de Estrategia de Procedimientos y Unidad de Estrategia de Problemas. Una vez seleccionado por el usuario el tipo de unidad que desea crear y los materiales que formaran parte de la misma, se creará dicha unidad, y se permitirá su visualización mediante el reproductor adecuado. El elemento que determina la composición de los materiales será el fichero `imsmanifest.xml`, que va a ser acorde con el estándar escogido, en este caso IMS Content Package.

Por tanto, el proyecto se enmarca dentro del área de e-learning, orientado al desarrollo de documentos didácticos por parte de educadores y especialistas en la educación.

1.2. Solución desarrollada

El nombre dado al sistema desarrollado ha sido **UolComposer**. Se trata de una aplicación WEB encaminada a facilitar el proceso de creación de unidades de aprendizaje interactivas que sigan el estándar IMS-LD [6]. Las principales características de la aplicación son:

- El usuario, una vez que ha realizado una búsqueda de los contenidos potencialmente interesantes para la unidad que quiere componer, deberá seleccionar aquellos recursos que quiera que formen parte de la misma. A su vez, debe seleccionar el tipo de unidad que va a crear, que puede ser de los tipos: Unidad de Aprendizaje basado en la realización de Problemas, Unidad de Aprendizaje basada en procedimientos y Unidad de Aprendizaje de Simple Colaboración.
- Cuando el usuario ha seleccionado todos estos datos, la aplicación realizará la composición de la misma, obteniéndose como resultado un paquete de contenidos que sigue la especificación IMS-LD, y que contiene todos los recursos necesarios para utilizar dicho paquete en actividades de aprendizaje.
- Si el usuario lo desea, puede publicar y visualizar la unidad de aprendizaje creada, haciendo uso para ello de la plataforma de software libre SLED, un reproductor de unidades de aprendizaje habitualmente utilizado.

El desarrollo de esta aplicación ha surgido debido a que la existencia de herramientas que realicen una composición de Paquetes de Contenidos están muy limitadas en cuanto a los usuarios que tienen acceso a las mismas, ya que se exige un conocimiento y dominio de la tecnología XML, que no está al alcance de la mayor parte de usuarios a los que se pretende dirigir este tipo de aplicación, que son principalmente profesores y gente del colectivo de la enseñanza que no suelen disponer de estos conocimientos, pero a los que sería de gran interés hacer uso de este tipo de herramienta. Es por ello que surge el proyecto de realización de esta aplicación WEB.

En el desarrollo del mismo se ha optado por hacer uso de la tecnología de Servlets+JSP del lenguaje JAVA, una tecnología potente para la realización de aplicaciones WEB que a su vez ofrece una gran escalabilidad si en un futuro se plantean cambios en la aplicación, como pueda ser modificar la presentación de la misma o añadir nuevas unidades de aprendizaje. Para esto también se ha hecho uso de algunos patrones de diseño como la Fábrica Abstracta (Abstract Factory) o los Identificadores, pero de esto ya se hablará en profundidad en otros apartados de esta Memoria.

La anotación con metadatos de cualquier documento es una labor tediosa que suele resultar agotadora, parte de confusa en un primer momento. Los objetos de aprendizaje, en su forma actual, vienen definidos por especificaciones y estándares de e-learning, que suelen incluir una implementación en forma de archivo XML. Para facilitar la creación de estos objetos de aprendizaje, se ha realizado esta aplicación, encaminada a ofrecer al personal educativo una herramienta que les permita crear de forma rápida, sencilla e intuitiva unidades de aprendizaje que sigan el estándar IMS LD, para su posterior publicación y utilización en procesos de aprendizaje.

El grado de configuración del sistema es relativamente alto, permitiendo al usuario elegir el tipo de unidad que quiere crear de entre los tres tipos existentes, así como la posterior publicación de la misma para ver el resultado de la composición realizada bajo sus premisas. Además, se han previsto como trabajos futuros la definición de nuevas áreas configurables del sistema.

2. Estado del Arte

2.1. El concepto e-learning

El aprendizaje electrónico, como lo podríamos traducir del vocablo inglés “electronic-Learning”, e-learning, es un tipo de enseñanza o capacitación que permite transmitir conocimientos, ya sea en tiempo real o asíncrono, a través de tecnología (Internet, Intranet o CD-ROM) para aprender de una manera más efectiva. La tecnología más utilizada para ello es Internet, dadas las posibilidades que ofrece de llevar la información a distintos lugares de manera rápida y efectiva a través de los ordenadores, algo que se ha venido incrementando desde que la red nos permite acceder a más personas y ofrecer ambientes de aprendizaje más elaborados.

Es posible encontrar varias definiciones distintas, pero si tuviéramos que dar una, podríamos definir el e-learning de la siguiente forma: *“El e-learning es todo aprendizaje, entrenamiento o educación que se facilita mediante el uso de las tecnologías telemáticas, especialmente las redes basadas en la tecnología de Internet”*. El e-learning es un método de educación a distancia, pero no toda educación a distancia es e-learning.

El modelo *e-learning* puro propugna un método de aprendizaje totalmente a distancia. No hay que confundirlo con el modelo híbrido (blended learning). Dicho modelo combina la enseñanza tradicional (asistencia a clases) con sesiones a distancia. En los inicios de desarrollo del e-learning se podría pensar que es un sustituto de la docencia presencial que hasta entonces había existido. El e-learning es una evolución del mismo que debería acabar por extinguirlo. La formación presencial puede ser combinada con un sistema de aprendizaje en línea que sirve a los alumnos para acceder en cualquier momento y en cualquier lugar a multitud de contenidos multimedia, herramientas de comunicación, etcétera. [7]

Una de las principales ventajas del e-learning es la facilidad de acceso. La formación puede llegar a más personas, puesto que desaparecen las barreras espacio-temporales. De esta forma, personas que antes tenían dificultades para acceder a la formación, ya sea debido a las dificultades de desplazamiento derivadas de vivir en pueblos o aldeas alejadas, o la falta de medios de transporte que se adapten a sus horarios, tiene ahora un abanico de posibilidades a su disposición para una formación continua. Es innegable que la enseñanza on-line ofrece una serie de ventajas:

- La flexibilidad horaria: el alumno accede en el momento en que tiene tiempo.
- La facilidad de acceso sin desplazamientos: se utilizan herramientas como correo electrónico, foros o chats para establecer comunicación entre los participantes.
- La formación personalizada.
- La actualización permanente de contenidos: por estar actualizado en la Web, el e-learning puede actualizarse instantáneamente, proporcionando la capacidad de perfeccionar el contenido más fácil y rápidamente y distribuir de forma inmediata la nueva información.
- El seguimiento exhaustivo del proceso formativo: los alumnos pueden comunicarse unos con otros, con el tutor y con los recursos on-line disponibles en Internet.
- La variedad de formatos (podcast, foros, entorno para el trabajo en grupo, materiales en formato Web, ejercicios interactivos, etc.)
- Aumenta el número de destinatarios: esta modalidad de formación se puede dirigir a una audiencia mucho más amplia.
- Aumenta la escalabilidad de los destinatarios: las soluciones de e-learning son altamente escalables. Los programas pueden pasar de 10 participantes a 100 o incluso 10.000 con un bajo coste.

Para dar soporte al aprendizaje a través de un ordenador, se han creado gran cantidad de herramientas que ofrecían diferentes servicios según las necesidades de cada organización. Por lo general, todos los sistemas siguen el esquema básico reflejado en la figura 1, que consiste en que el sistema ofrece interfaces para crear contenidos, para controlar y gestionar la utilización de los mismos y por último otro para administrar el sistema (alta de usuarios, baja de usuarios, etc.). Además ofrece otro interfaz más al alumno, que será el utilizado para el proceso de aprendizaje en sí.

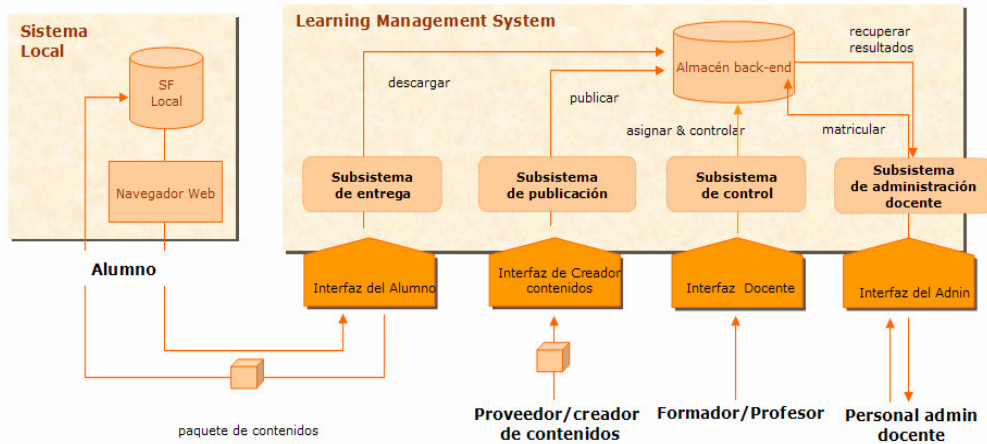


Figura 1: Esquema básico de funcionamiento de un sistema de e-learning

El objetivo de utilizar esta arquitectura es separar el contenido de la presentación, facilitar los flujos de trabajo alumno/alumno y alumno/profesor, otorgar el acceso a ciertos contenidos, controlar las versiones de todo el material, así como temporizar su publicación.

Dichas herramientas de soporte al aprendizaje se denominan Learning Management Systems (LMS) o Gestores de contenidos educativos, que permiten planificar, implementar y evaluar procesos de aprendizaje. Estos sistemas suelen permitir la autoría, navegación y agregación de contenidos. Un LMS registra usuarios, organiza los diferentes cursos, almacena datos sobre los usuarios o también provee informes para la gestión. Además suministra al profesor un mecanismo para distribuir contenido, supervisar la participación de los estudiantes y evaluar su desempeño. También suelen ofrecer a los estudiantes el uso de mecanismos de interacción como foros de discusión o chats. Un LMS, por lo general, incluye posibilidades de autoría (crear sus propios contenidos), pero se centra en gestionar contenidos creados por una gran variedad de fuentes diferentes. Los LMS más extendidos son WebCT [8] y Blackboard [9], que son comerciales, y Moodle [10] y dotLRN [11] que son Open Source, aunque en realidad existen decenas de ellos.

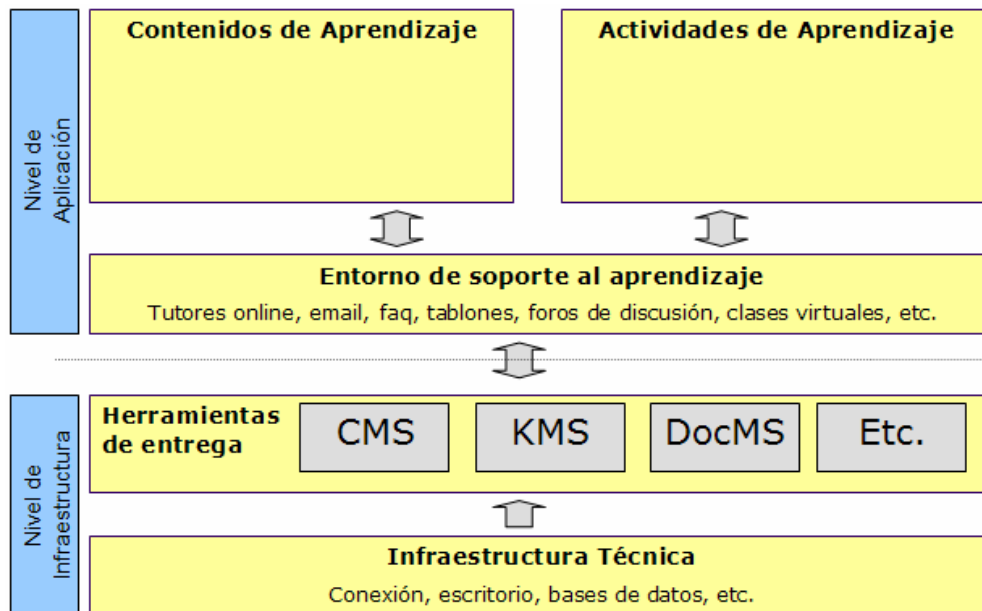


Figura 2: Esquema de aplicaciones de soporte al e-learning

En general, la existencia de tal cantidad de sistemas supone un inconveniente a la hora de reutilizar los contenidos de manera eficaz, es por esto que surge la necesidad de la utilización de algún lenguaje de Metadatos estandarizado que permita la descripción de los objetos de aprendizaje de manera que pueda ser utilizado en cualquier plataforma. Es en este punto donde entran los estándares de metadatos, cómo pueden ser el IMS LRMI o el IEEE LOM.

2.2. Estandarización

La tecnología e-learning es una tecnología relativamente joven, su historia únicamente abarca sólo unos pocos años. Es por ello que cuando una tecnología empieza a dar sus primeros pasos todo el mundo tiene sus propios criterios a la hora de hacer las cosas. La falta de criterios comunes termina creando mucha confusión y eso es precisamente lo que ha pasado con la tecnología e-learning. Es por esto que la estandarización resulta un objetivo fundamental y necesario para el crecimiento y asentamiento de la tecnología e-learning. Dicha estandarización de la metodología de productos de e-learning debe realizarse tanto en contenido como en infraestructura, y debe garantizar una serie de objetivos: Accesibilidad, Interoperabilidad, Durabilidad y Reutilización.

Un estándar e-learning sería el vehículo a través del cual sería posible dotar de flexibilidad a las soluciones e-learning, empaquetándose de una manera más coherente los recursos y los contenidos tanto para los desarrolladores como para los estudiantes. Esto es muy importante ya que todo producto que se adhiriera a los estándares no quedará obsoleto a corto plazo, logrando de esta manera que queden protegidas así las inversiones realizadas en este tipo de productos. Y por supuesto, la estandarización de los productos de e-learning es un requisito imprescindible para el éxito de la economía del conocimiento y para el futuro desarrollo de la tecnología e-learning.

En el mercado existen tanto los LMS (Learning Management System, que son los gestores del producto de e-learning a nivel de usuarios, de cursos y de comunicaciones) como los Courseware (contenidos de los cursos) lo que hace imprescindible una normativa que compatibilice los diferentes sistemas y cursos a fin de que un curso de cualquier fabricante pueda ser cargado en cualquier LMS de otro fabricante, y que los resultados de la actividad de los usuarios en el curso puedan ser registrados por el LMS.

Los diferentes estándares que se desarrollan hoy en día para la industria del e-learning se pueden clasificar en los siguientes tipos [12]:

1. Sobre el contenido o el curso:

- Estructuras de contenidos.
- Empaquetamiento de contenidos.
- Seguimientos de resultados.

2. Sobre el alumno:

- Almacenamiento e intercambio de información del alumno.
- Habilidades del alumno.
- Privacidad y seguridad.

3. Sobre la interoperabilidad:

- Integración de componentes del LMS.
- Interoperabilidad entre múltiples LMS.



Figura 3: Esquema de actores de un sistema de e-learning

Al hablar sobre un estándar e-learning nos estamos refiriendo a un conjunto de reglas en común para las compañías dedicadas a la tecnología del e-learning. Estas reglas proveen unos modelos comunes de la información para cursos e-learning y plataformas LMS, los cuáles, básicamente, permiten a los sistemas y a los cursos compartir datos o “hablar” con otros. Esto también nos da la posibilidad de incorporar contenidos de distintos proveedores en un solo programa de estudios. Estas reglas, además, definen un modelo de empaquetamiento estándar para los contenidos. Los contenidos pueden ser empaquetados como “objetos de aprendizaje” (learning objects o LO), de tal forma que permitan a los desarrolladores crear contenidos que puedan ser fácilmente reutilizados e integrados en distintos cursos.

Finalmente, los estándares permiten crear tecnologías de aprendizaje más poderosas, y “personalizar” el aprendizaje basándose en las necesidades individuales de los diferentes alumnos que tomen parte en el curso. Básicamente, lo que se persigue con la aplicación de un estándar para el e-learning es lo siguiente:

1. Durabilidad: que la tecnología desarrollada con el estándar sea duradera y que evite que, de forma rápida, los cursos se vuelvan obsoletos.

2. Interoperabilidad: que sea posible intercambiar información entre los diferentes LMS.

3. Accesibilidad: que se permita un seguimiento de los progresos de los alumnos.

4. Reusabilidad: que los diferentes cursos y objetos de aprendizaje puedan ser reutilizados con diferentes herramientas y en distintas plataformas, puesto que lo contrario supondría depender exclusivamente de los contenidos producidos “in-company” o bien de los contenidos de que disponga el fabricante.

Estrictamente hablando, no existe un estándar e-learning disponible hoy en día. Lo que existe es un conjunto de grupos y organizaciones que desarrollan especificaciones (protocolos). Estas especificaciones no dejan de ser recomendaciones que, por el momento, la industria trata de seguir.

En los años siguientes, el trabajo de las distintas organizaciones que están trabajando en las especificaciones para estándares e-learning estará centrado en los siguientes temas:

- Repositorio de Contenidos

Las organizaciones se están focalizando fuertemente en estándares de contenidos e-learning. El principal objetivo es tener repositorios de objetos de aprendizaje (learning objects) reusables, de tal manera que puedan ser montadas en unidades de aprendizaje adaptativas y entregadas por cualquier plataforma e-learning. Sin embargo, uno de los mayores problemas que enfrenta hoy en día la industria del e-learning es la interoperabilidad de los contenidos de aprendizaje.

- Internacionalización y Localización

Los distintos grupos que están desarrollando especificaciones para e-learning participan en forma activa en todo el mundo y cada día existe una mayor colaboración entre ellas. Esto genera dos desafíos: la creación de estándares “culturalmente” neutrales (internacionalización), y la adaptación de los estándares a las necesidades locales (localización).

- Programas de certificación

Existe un creciente énfasis en crear test de compatibilidad y programas de certificación. ADL está trabajando en un programa de certificación. Actualmente sólo existen programas de certificación para AICC.

- Arquitectura

La industria del e-learning ha estado creciendo sin tener una clara visión de los componentes de un sistema e-learning y de la forma en que interactúan. La necesidad de definir una arquitectura global es crítica para la evolución del desarrollo de estándares.

2.3. Iniciativas de Estándares de e-learning

Tener una idea clara del proceso de estandarización para las tecnologías e-learning es un atarea compleja, debido al relativamente poco tiempo que se lleva realizando el proceso y a la profusión de grupos, instituciones y consorcios que trabajan en el tema. No obstante, el escenario está mejorando, ya que cada vez más se llegan a acuerdos de colaboración ente distintas iniciativas. Por tanto se está cada vez más cerca de una estandarización real y que tenga un impacto en la industria.

Dentro de las principales iniciativas de estándar para e-learning podemos mencionar:

2.3.1. AICC, Aviation Industry CBT Committe

Fue el primer organismo creado para crear un conjunto de normas que permitiese el intercambio de cursos CBT entre diferentes sistemas.

2.3.2. IEEE Learning Technologies Standards Committee (LTSC)

Se trata de un organismo que promueve la creación de una norma ISO, una normativa estándar real de amplia aceptación. El LTSC se encarga de preparar normas técnicas, prácticas y guías recomendadas para el uso informático de componentes y sistemas de educación y de formación, en concreto, los componentes de software, las herramientas, las tecnologías y los métodos de diseño que facilitan su desarrollo, despliegue, mantenimiento e interoperación.

Lo que hizo fue recoger el trabajo del comité de la AICC y mejorarlo, creando la noción de metadato (información sobre los datos, una descripción más detallada que la ofrecida por la AGR 010 de la AICC de los contenidos del curso).

LTSC tiene más de una docena de grupos de trabajo (working groups o WGs) y grupos de estudio (study groups o SGs) que desarrollan especificaciones para la industria del e-learning.

2.3.3. IMS Global Learning Consortium, Inc.

Este Consorcio está formado por miembros provenientes de organizaciones educacionales, empresas públicas y privadas. Su misión es desarrollar y promover especificaciones abiertas para facilitar las actividades del aprendizaje on line. El trabajo de la IEEE fue recogido por esta corporación privada creada por algunas de las empresas más importantes del sector. Su objetivo fue la creación de un formato que pudiese en práctica las recomendaciones de la IEEE y la AICC.

Lo que se hizo fue definir un tipo de fichero XML para la descripción de los contenidos de los cursos. De tal modo que cualquier LMS pueda, leyendo su fichero de configuración IMSMANIFEST.XML, cargar el curso.

2.3.3.1. Especificaciones de IMS

El objetivo de IMS de definir especificaciones que hagan posible la interoperabilidad de aplicaciones y servicios de enseñanza distribuida, se ha concretado, a día de hoy, en más de 15 especificaciones principales.

A continuación se describen las principales iniciativas de este comité:

1. Learning Object Metadata (LOM)

Esta especificación entrega una guía sobre cómo los contenidos deben ser identificados o “etiquetados” y sobre cómo se debe organizar la información de los alumnos de manera de que se puedan intercambiar entre los distintos servicios involucrados en un sistema de gestión de aprendizaje (LMS). La especificación para metadata del IMS consta de tres documentos:

IMS Learning Resource Meta-data Information Model, IMS Learning Resource XML Binding Specifications, IMS Learning Resource Meta-data Best Practices and Implementation Guide.

2. Empaquetamiento de Contenidos (Content Packaging)

Esta especificación provee la funcionalidad para describir y empaquetar material de aprendizaje, ya sea un curso individual o una colección de cursos, en paquetes portables e ínter operables. El empaquetamiento de contenidos está vinculado a la descripción, estructura, y ubicación de los materiales de aprendizaje on-line, y a la definición de algunos tipos particulares de contenidos.

La idea es que el contenido desarrollado bajo este estándar sea utilizado en una variedad de sistemas de gestión de aprendizaje (LMS). Esta especificación ha sido comercializada por Microsoft bajo el nombre de LRN (Learning Resource Interchange).

3. Interoperabilidad de Preguntas y Tests (Question and Test Interoperability, QTI)

El IMS QTI propone una estructura de datos XML para codificar preguntas y test online. El objetivo de esta especificación es permitir el intercambio de estos tests y datos de evaluación entre distintos LMS.

4. Empaquetamiento de Información del Alumno (Learner Information Packaging, LIP)

Esta especificación define estructuras XML para el intercambio de información de los alumnos entre sistemas de gestión de aprendizaje, sistemas de recursos humanos, sistemas de gestión del conocimiento, y cualquier otro sistema utilizado en el proceso de aprendizaje. Actualmente, existen varios desarrolladores de productos que tienen en vista adoptar esta especificación.

5. Secuencia Simple (Simple Sequencing)

Esta especificación define reglas que describen el flujo de instrucciones a través del contenido según el resultado de las interacciones de un alumno con el contenido. Esta representación de flujo condicionado puede ser creada manualmente o a través de herramientas compatibles con esta especificación. Una vez creado, la representación de la secuencia puede ser intercambiada entre sistemas diseñados para entregar componentes instruccionales a los alumnos.

6. Diseño del Aprendizaje (Learning Design)

Este grupo de trabajo del IMS investiga sobre las maneras de describir y codificar las metodologías de aprendizaje incorporadas en una solución e-learning.

7. Repositorios Digitales (Digital Repositories)

El IMS está en el proceso de creación de especificaciones y recomendaciones para la inter-operación entre repositorios digitales.

8. Definición de competencias (Competency Definitions)

El IMS (al igual que la IEEE) está en el proceso de crear una manera estandarizada de describir, referenciar e intercambiar definiciones de competencias. En esta especificación, el término competencia es usado en un sentido muy general, que incluye habilidades, conocimiento, tareas, y resultados de aprendizaje. Esta especificación entrega una manera de representar formalmente las características principales de una competencia, independiente de su uso en un contexto en particular, permitiendo así su interoperabilidad entre distintos LMS.

9. Accesibilidad (Accessibility)

Este grupo de trabajo promueve el contenido de aprendizaje accesible a través de recomendaciones, guidelines, y modificaciones a otras especificaciones. Tecnología accesible se refiere a la tecnología que puede ser usada sin tener acceso pleno a uno o más canales de entrada y salida, usualmente visuales y auditivos.

2.3.3.1.1. Estructura de las especificaciones de IMS

Normalmente cada una de ellas se encuentra detallada al menos en tres documentos:

- **Guía de Implementación y consejos.** En él se incluyen: la forma de uso de la especificación, ejemplos, la relación con otras especificaciones, y cualquier tipo de información complementaria que pueda servir de ayuda. Normalmente es el documento que se recomienda leer primero para entender los conceptos generales con los que se trata.
- **Modelo de Información.** Documento que describe de manera formal, los datos así como su estructuración, detallando cada uno de los elementos considerados en la especificación. El modelo que se propone en este documento es independiente del formato físico en el que finalmente se representa la información.
- **Documento de Enlace.** Documento que ofrece la forma de representar la estructura de datos de la especificación, generalmente, en XML. Adicionalmente se proporciona el esquema documental XML que nos permite comprobar la validez de la estructura de un documento que hayamos creado, respecto a la especificación a la que está asociado.

IMS tiene muchas especificaciones ya que cada una de ellas está enfocada en una necesidad distinta del proceso de enseñanza. A continuación se va a detallar en profundidad el IMS Learning Design, que es la especificación que ha sido incluida y tratada en este proyecto en concreto.

IMS Learning Design

Esta especificación ha sido el resultado de la integración dentro de IMS de la especificación Educational Modeling Language (lenguaje de modelado educacional) (Koper 2001), desarrollada inicialmente en la Universidad Abierta de Holanda. Se ocupa de describir y codificar el diseño pedagógico, es decir las metodologías educativas implícitas en un proceso de enseñanza, de forma que sean procesables por un LMS. En este caso se utiliza un nuevo concepto, la unidad de aprendizaje (UdA), ya que se considera que lo importante no son tanto los objetos de aprendizaje por sí mismos, si no las actividades en las que se encuentran implicados. El elemento clave de una Unidad de Aprendizaje es la actividad o tarea, que se concibe como uno o más actores (p.e. alumnos, profesores) que trabajan para lograr un cierto objetivo educativo en un determinado entorno. El entorno contiene los recursos y los servicios necesarios para realizar la actividad propuesta. El principio subyacente es que los alumnos aprenden realizando actividades en un entorno, en el cual los objetos de aprendizaje son recursos que permiten o facilitan la tarea. La visión es más amplia que la de los objetos de aprendizaje básicos, ya que se contempla el uso de herramientas o de procesos, como la comunicación entre alumnos o entre alumnos y profesores. De hecho el rol o papel de un alumno podría cambiar en un determinado momento, por ejemplo, para supervisar el trabajo realizado por otros alumnos. La unidad de aprendizaje es la nueva unidad mínima de intercambio entre sistemas, ya que se considera que si se descompone en sus elementos básicos se pierde el diseño pedagógico que permite alcanzar el resultado deseado.

La estructura de un paquete de contenidos es:

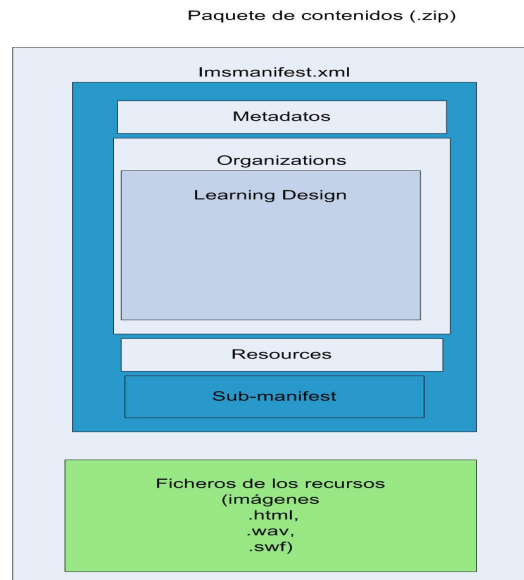


Figura 4: Esquema de la estructura de un paquete de contenidos del estándar IMS Content Package

Un paquete de contenidos es un fichero en formato .zip. Dicho fichero está compuesto por diferentes elementos, todos ellos necesarios para formar el paquete de contenidos referente a un tema en particular en función de lo que su autor considere necesario. La parte más importante de un paquete de contenidos es el fichero de manifiesto (imsmanifest.xml). Dicho fichero contiene la definición de todo el paquete de contenidos para que éste pueda ser utilizado por los usuarios, pudiendo reproducirlo y visualizarlo tantas veces como sean necesarias hasta que haya adquirido los conocimientos del mismo. Por lo tanto es el elemento fundamental de un paquete de contenidos, sin él no sería posible realizar nada. Se pueden diferenciar las partes fundamentales de un fichero de manifiesto:

- **Metadatos:** es un conjunto de información que permite identificar el paquete de contenidos en un contexto en el que este con otros paquetes de contenidos. Estos datos contienen información acerca del tipo de contenidos que contienen los paquetes, así como de su autor y su duración.
- **Organizations:** este elemento contiene las diferentes agrupaciones de elementos que componen el paquete de contenidos. Dentro del mismo, se encuentra la unidad de aprendizaje, la cuál está formada por:
 - **Componentes:** es la parte más importante de la unidad de aprendizaje. En esta parte se define el “que” de una unidad, es decir, las partes que contiene, Las más importantes son:
 - **Roles:** contiene los distintos roles que se pueden asumir en el contexto de la unidad, son de dos tipos fundamentales, Profesor y Alumno.
 - **Actividades:** estos elementos son las distintas actividades que un alumno va a poder realizar en un curso. Cada actividad contiene un nombre, un título, y una asociación con los recursos que necesita para mostrarse y que permita que el alumno la realice. Cada unidad de aprendizaje puede tener varias actividades, todas ellas declaradas en esta parte del paquete de contenidos. Existe un elemento denominado estructura de actividades, que como su propio nombre indica, consiste en un agregado de actividades para formar una única unidad.
 - **Environments:** estos elementos constituyen la relación de las actividades con cada uno de los recursos que va a necesitar la actividad para realizarse. Cada environment está formado por un título y un objeto de aprendizaje, que es un elemento que se asocia con un recurso y que permite referenciar a archivos (.zip, .html....) en los que está la información real de la unidad de aprendizaje.
- **Métodos:** esta parte del paquete de contenidos indica la forma en que se va a mostrar al usuario el paquete al ser reproducido por un visualizador de unidades de aprendizaje. Esta compuesto por visualizaciones que contienen

actos, los cuáles se asocian con las distintas actividades y estructura de actividades que conforman la unidad. Es gracias a estos elementos que se va a poder definir el orden los elementos de aprendizaje, así como las condiciones de terminación de cada uno de las actividades del paquete para poder pasar a la siguiente actividad.

- **Recursos:** esta parte del paquete de contenidos contiene la declaración de todos los archivos físicos en los que va a ir el contenido real del paquete, es decir, los archivos de diferentes formatos (.doc, .pdf, .zip.....) en los que se puede encontrar la información de todos y cada uno de los temas tratados por la unidad de aprendizaje. Puede incluir tantas entradas como sea necesario, para que una unidad puede tener todos los recursos que el creador de la misma considere oportuno.

En el paquete de contenidos se incluyen también, como es natural, los ficheros físicos con la información didáctica de las actividades del paquete de contenidos, es decir, los ficheros en formato Word, pdf, wav... que se van a visualizar en el reproductor de paquetes de contenidos, y que van a resultar imprescindibles para mantener la integridad de paquete. La estructura del fichero de manifiesto es definida por el siguiente modelo:

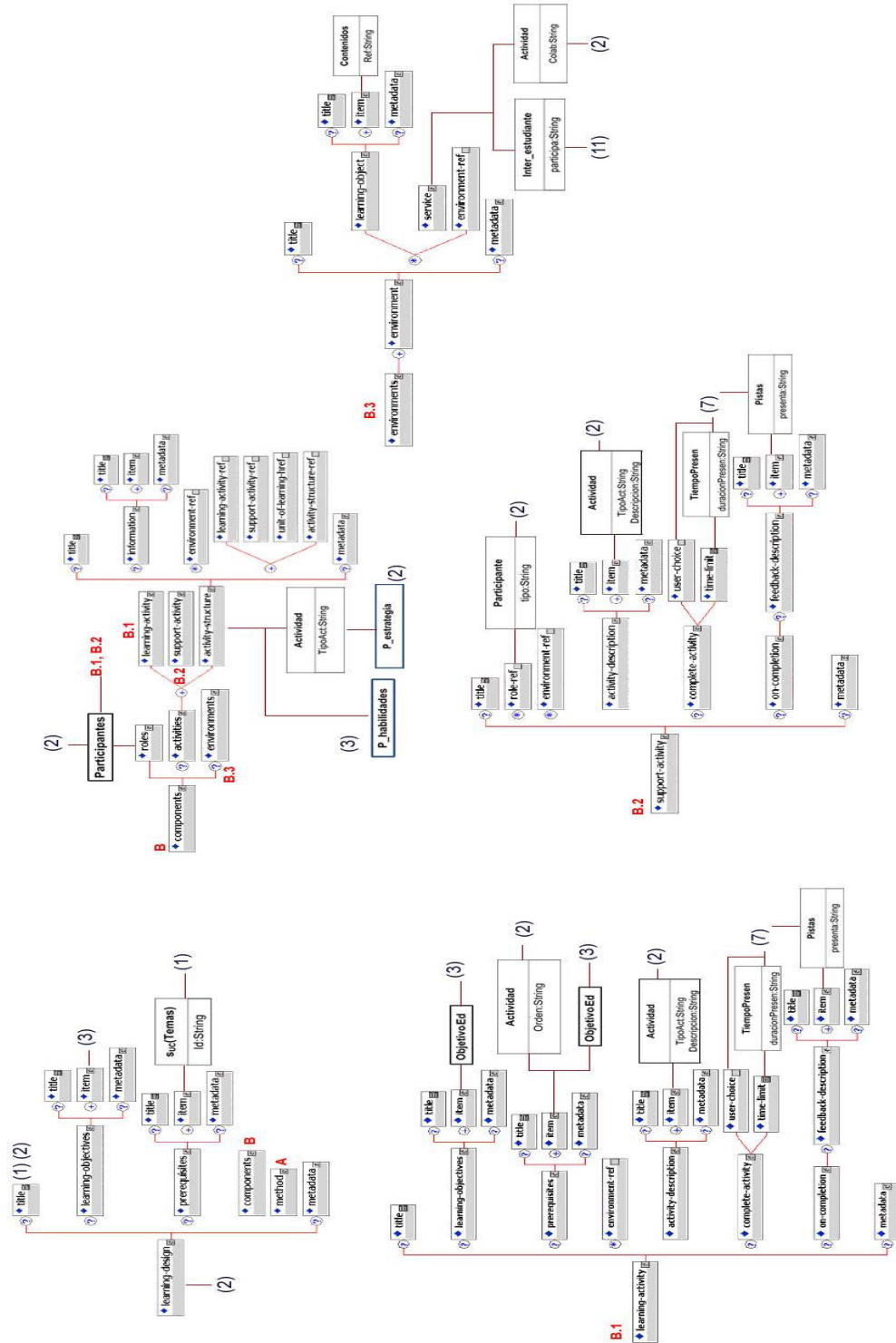


Figura 5: Estructura de un fichero de manifiesto basada en IMS Learning Design según MD2 (parte 1).

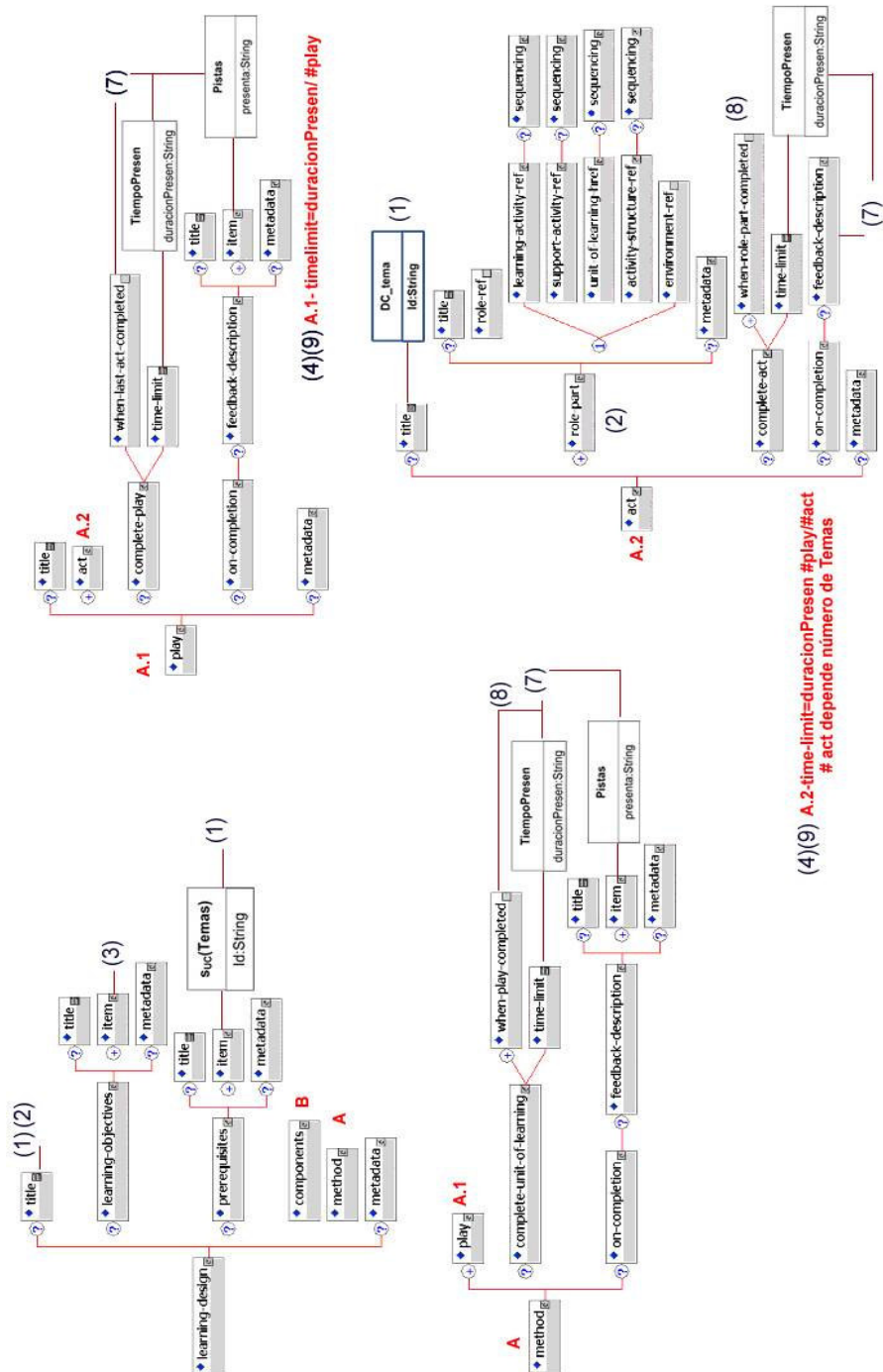


Figura 6: Estructura de un fichero de manifiesto de IMS Learning Design según MD2 (parte 2)

Como ya hemos visto, la estructura de los paquetes de contenidos es común. Pero a partir de esa estructura, existen diferentes tipos de unidades específicas. A continuación se detallan las tres que se han utilizado en este proyecto, que son las tres principales existentes en la actualidad, pero en un futuro próximo existe la posibilidad de que aparezcan nuevos tipos de unidades, con lo que se ha de tener en cuenta esa posibilidad en la implementación del sistema. Pero eso se detallará más adelante en este documento. Los tres tipos principales de unidades son:

1. Unidad de aprendizaje basado en problemas:

La estrategia pedagógica de esta unidad es el aprendizaje basado en problemas. En esta estrategia, el aprendizaje de los estudiantes se basa en la resolución de un determinado problema relacionado con una temática específica.

El problema se puede descomponer en pequeños problemas de manera que el estudiante vaya alcanzando progresivamente las habilidades, competencias y conocimientos relacionados con la temática.

Los participantes en este tipo de aprendizaje son los estudiantes.

2. Unidad de aprendizaje basado en procedimientos:

La estrategia pedagógica de esta unidad es el aprendizaje basado en procedimientos. El objetivo es llegar a dominar conceptos, procesos y procedimientos, generalmente estables y relacionados con una temática determinada.

En esta estrategia el aprendizaje de los estudiantes se basa en el estudio de ciertos contenidos sobre las definiciones de los conceptos, procedimientos y procesos relacionados con una determinada temática.

Los participantes en este tipo de aprendizaje son los estudiantes.

3. Unidad de aprendizaje colaborativo:

La estrategia pedagógica de esta unidad es el aprendizaje colaborativo. El funcionamiento del aprendizaje con esta estrategia consiste en la interacción entre los estudiantes para construir y compartir conocimientos, habilidades y competencias.

Esta es la implementación más sencilla de este tipo de estrategia, en la que sólo se debate sobre un tema determinado.

Los participantes en este tipo de aprendizaje son los estudiantes y los profesores.

2.3.4. ADL SCORM

Formada en 1997, la iniciativa ADL (Advanced Distributed Learning), es un programa del Departamento de Defensa de los Estados Unidos y de la Oficina de Ciencia y Tecnología de la Casa Blanca para desarrollar principios y guías de trabajo necesarias para el desarrollo y la implementación eficiente, efectiva y en gran escala, de formación educativa sobre nuevas tecnologías Web.

Las especificaciones de SCORM están organizadas como “libros” separados. La mayoría de estas especificaciones son tomadas desde otras organizaciones. Estos “libros” técnicos se agrupan bajo dos tópicos principales: Content Aggregation Model y Run-Time Environment.

Como lo muestra la siguiente figura, la actual versión 1.2 de SCORM ha sido dividida en tres libros que se detallan a continuación:

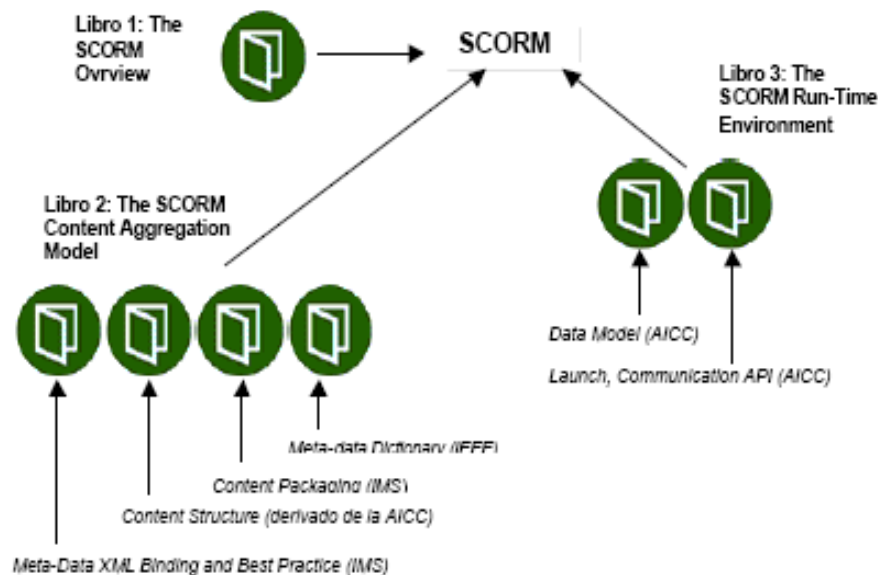


Figura 7: División en libros de la actual versión de SCORM

- **Libro 1: Scorm Overview.** Contiene una descripción general de la iniciativa de ADL, un análisis de SCORM, y un resumen de las especificaciones técnicas contenidas en las siguientes secciones.

- **Libro 2: Scorm Content Aggregation Model.** Contiene una guía para identificar y agregar recursos dentro de un contenido de aprendizaje estructurado. Este libro describe una nomenclatura para el contenido de aprendizaje, describe el SCORM Content Packaging (empaquetamiento de contenidos) y hace referencia al IMS Learning Resource Meta-data Information Model, el cual está basado en el IEEE LTSC Learning Object Metadata (LOM) Specification, que fue el resultado de un esfuerzo en conjunto entre el IMS Global Learning Consortium y la Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE).

- **Libro 3: Scorm Run-Time Environment.** Incluye una guía para lanzar contenidos y hacerle un seguimiento en un ambiente basado en Web. Este libro es derivado del CMI001 Guidelines for Interoperability de la AICC.

SCORM también divide el mundo de la tecnología e-learning en componentes funcionales. Los principales componentes son: Learning Management System (LMS) y Sharable Content Objects (SCOs). SCO se refiere a objetos de aprendizaje reusables y estandarizados. Otros componentes en el modelo SCORM son herramientas que crean los SCOs y los ensamblan en unidades de aprendizaje más grandes (un curso por ejemplo).

2.4. Plataformas de enseñanza (LMS)

Es el núcleo alrededor del cual giran los demás elementos. Básicamente se trata de un software para servidores de Internet/Intranet que se ocupa de:

- Gestionar los usuarios: inscripción, control de sus aprendizajes e historial, generación de informes, etc.
- Gestionar y lanzar los cursos, realizando un registro de la actividad del usuario: tanto los resultados de los tests y evaluaciones que realice, como de los tiempos y accesos al material formativo.
- Gestionar los servicios de comunicación que son el apoyo al material online, foros de discusión, charlas, videoconferencia; programarlos y ofrecerlos conforme sean necesarios.

El panorama actual de los LMS está caracterizado por su gran dispersión, ya que todavía no hay entre ellas ningún liderazgo claro comparable al existente en otras áreas de software, como por ejemplo en los programas de ofimática: procesadores de texto, hojas de cálculo, etc. Pero es necesario destacar, entre todos ellos, el Moodle, el cuál detallaremos a continuación con carácter introductoria para dar los datos más relevantes del mismo:

2.4.1. Moodle

Moodle es una plataforma de e-learning libre, a este tipo de sistemas también se le denomina Ambiente de Aprendizaje Virtual - AVA o plataforma para la educación en línea. Moodle es un Sistema de Gestión de Enseñanza (Course Management System - CMS o Learning Management System - LMS), es decir una aplicación diseñada para ayudar a los profesores a crear cursos en línea. La ventaja de esta plataforma respecto a otras es que está construido sobre la base de la pedagogía social constructivista.

Moodle es un proyecto en desarrollo, diseñado para crear y ofrecer cursos en línea desde la perspectiva del constructivismo social, con el propósito de ofrecer una alternativa libre y de fácil uso a instituciones y personas en el ámbito educativo que quieren hacer un mejor uso de Internet.

El Proyecto Moodle es Libre (desde la perspectiva del Software Libre) porque sus desarrolladores tienen la firme convicción de la importancia de una educación sin restricciones y para todos, y en este sentido Moodle pretende ser un medio para contribuir a la realización de estos ideales.

Historia de Moodle

El Proyecto Moodle (Modular Object-Oriented Dynamic Learning Environment o Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular) fue iniciado por Martin Dougiamas (Australia), quien trabajó como administrador de WebCT en la Universidad de Curtin, y decepcionado por la complejidad de esta plataforma pues no era tan intuitiva para los usuarios, lo llevaron a hacer una Maestría y un Doctorado en Educación que combinándola con su carrera de informática (Ciencias de la Computación) e influenciado por el constructivismo en la pedagogía, dio como resultado a Moodle en 1999, una herramienta intuitiva y fácil de usar.

Actualmente colaboran en el proyecto alrededor de 100 personas entre desarrolladores (cerca de 50), traductores (otros 40), beta-testers entre otros. Moodle ha venido evolucionando desde 1999 y nuevas versiones siguen siendo producidas.

En el 2003 se presentó **moodle.com** una empresa que ofrece soporte comercial adicional para aquellos que lo necesitan mediante los Moodle Partners (grupo de empresas de servicios) que ofrecen una amplia oferta de servicios comerciales para los usuarios, entre los cuales se tiene hosting de Moodle completo, contratos de soporte remoto, desarrollos a medida y consultoría. Estos Partners están supervisados por Dougiamas y su equipo.

Características de Moodle

Moodle es un proyecto activo y en constante evolución, que posee muchas características, entre las que se tienen:

- Promueve una pedagogía constructivista social.
- Adecuada para el 100% de las clases en línea.
- Puede especificarse la fecha final de entrega de una tarea y la calificación máxima que se le podrá asignar.
- Admite la presentación de cualquier contenido digital, Word, PowerPoint, Flash, vídeo, sonidos, etc.
- Los estudiantes pueden subir sus tareas (en cualquier formato de archivo) al servidor. Se registra la fecha en que se han subido.

- En lo referente a los talleres permite la evaluación de documentos entre iguales, y el profesor puede gestionar y calificar la evaluación.
- Hay diferentes tipos de foros disponibles: exclusivos para los profesores, de noticias del curso y abiertos a todos.

3. Planteamiento del Problema

3.1. Herramientas de autoría

3.1.1. RELOAD

Para la implementación de la aplicación WEB, se realizó un primer acercamiento a una de las herramientas de edición de unidades de E-learning ya existente, el RELOAD. Es una herramienta multiplataforma, desarrollada en java, para la creación de objetos de aprendizaje reutilizables. Entre sus herramientas, encontrarás un editor, un reproductor de objetos reutilizables y aplicaciones de diseño de acciones formativas. Se trata de crear objetos de aprendizaje que cumplan la estandarización SCORM, o la especificación IMS, en aras de poder insertarlos en las plataformas de e-Learning más utilizadas (Moodle, WebCT, Blackboard, Atutor,...).

El RELOAD es una herramienta que está compuesta fundamentalmente por tres paneles: el panel de recursos(a la izquierda de la pantalla), el panel del manifiesto(a la derecha) y un panel de atributos (abajo). El panel del manifiesto es la parte principal de la aplicación, dado que es en donde se ve el contenido y la estructura de la unidad que está creando el usuario, con el Manifiesto que contiene los metadatos, los recursos y las organizaciones. El panel de atributos contiene diferentes opciones en forma de selectores y valores de los diferentes elementos de la unidad.

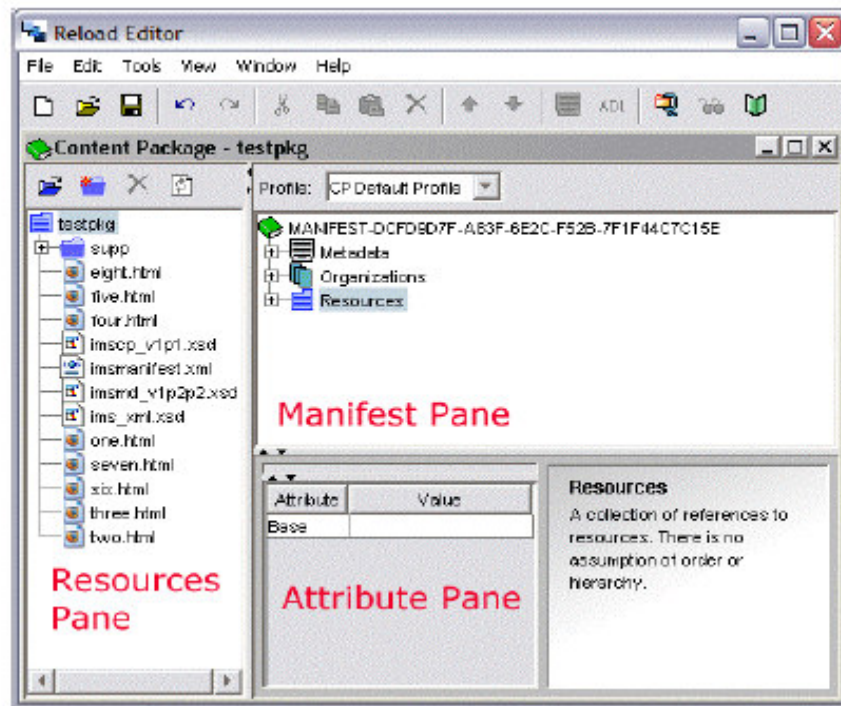


Figura 8: Herramienta RELOAD (parte 1)

La aplicación permite que se abran distintas ventanas en las que ver los contenidos del manifiesto, como pueden ser los metadatos, todo ello como ya se ha comentado se abrirá en una ventana auxiliar de la forma de la siguiente:

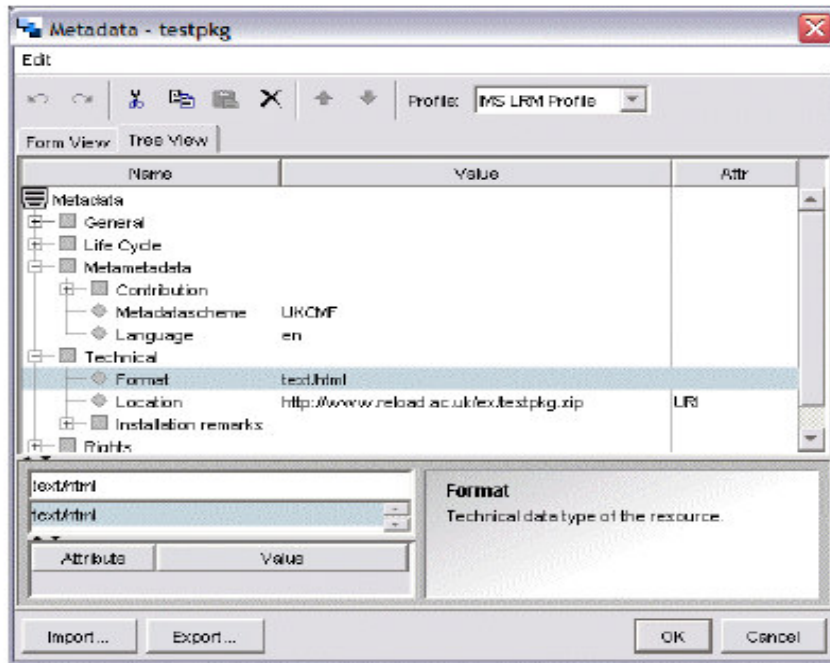


Figura 9: Herramienta RELOAD (parte 2)

En la parte superior se pueden observar las clásicas funciones de los editores, en una barra de herramientas, opciones tales como: crear una nueva unidad, cortar, copiar, pegar.... y otras mucho más específicas del contexto de la aplicación, esto es, crear un paquete de contenidos, importar recursos...

Funciones principales de RELOAD

La principal función sobre la que vamos a centrar nuestra atención es la de creación de una unidad de aprendizaje con el RELOAD. A la hora de empezar, vemos que lo primer que nos pide es fijar la ruta en la que crear el paquete de contenidos. Una vez seleccionado, nos aparece la pantalla principal del editor, en la que vemos los tres paneles ya comentados antes, el panel de recursos, el panel del manifiesto y el panel de atributos. Cuando se crea un paquete de datos, lo primero que el RELOAD crea automáticamente es el `imsmanifest.xml`, que es el fichero central y descriptor del paquete que se va a crear. También son creados los schema del paquete, necesarios para su publicación. Si queremos añadir metadatos, hemos de pulsar en la opción que nos permite hacerlo, y aquí nos encontramos el principal problema del RELOAD: se exige un conocimiento exhaustivo de XML. Un principiante va a encontrar totalmente

imposible crear metadatos, puesto que no entenderá nada de lo que se le pide que introduzca, con lo que las funcionalidades del RELOAD se ven severamente podadas por tal inconveniente. De todos modos, con un aprendizaje de XML bastante avanzado se podrá seguir utilizando el editor, que es lo que se va a seguir haciendo en este apartado.

Para poder crear el paquete de contenidos, es necesario importar los contenidos que lo van a formar. Para ello, hemos de pulsar en la opción de “Import Content”, la cuál nos abre una ventana de exploración de carpetas en la que tenemos que seleccionar los archivos. El RELOAD nos permite, marcando una opción predefinida, importar los recursos asociados (imágenes, hojas de estilo, etc) a los HTML, algo que es bastante útil. Ya habremos seleccionado la carpeta raíz del paquete de contenidos, donde quedaran importados todos los que seleccionemos, para poder seguir trabajando con ellos. La pantalla queda con una perspectiva similar a esta:

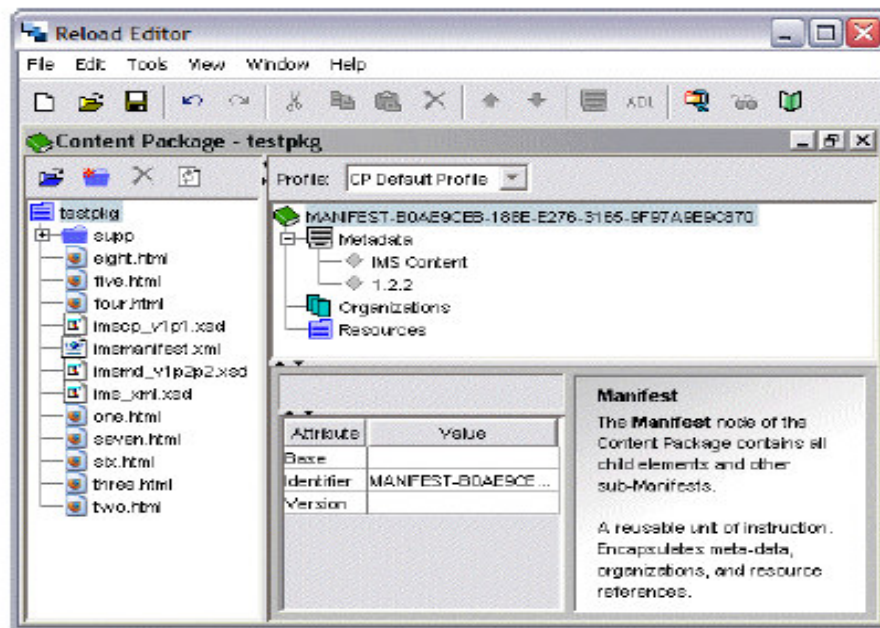


Figura 10: Herramienta RELOAD (parte 3)

La siguiente actividad a realizar es crear las organizaciones que formaran parte del manifiesto. Un manifiesto va a contener una o más organizaciones. Para añadirla, hay que seleccionar con el botón derecho el nodo de organizaciones y seleccionar “Add Organization”. Veremos de nuevo una ventana en la que hemos de incluir valores para los que es necesario un conocimiento de XML avanzado. Una vez introducidos, si se conoce el XML, se puede poner un título a la organización, el que se desee.

El aspecto de la pantalla principal es:

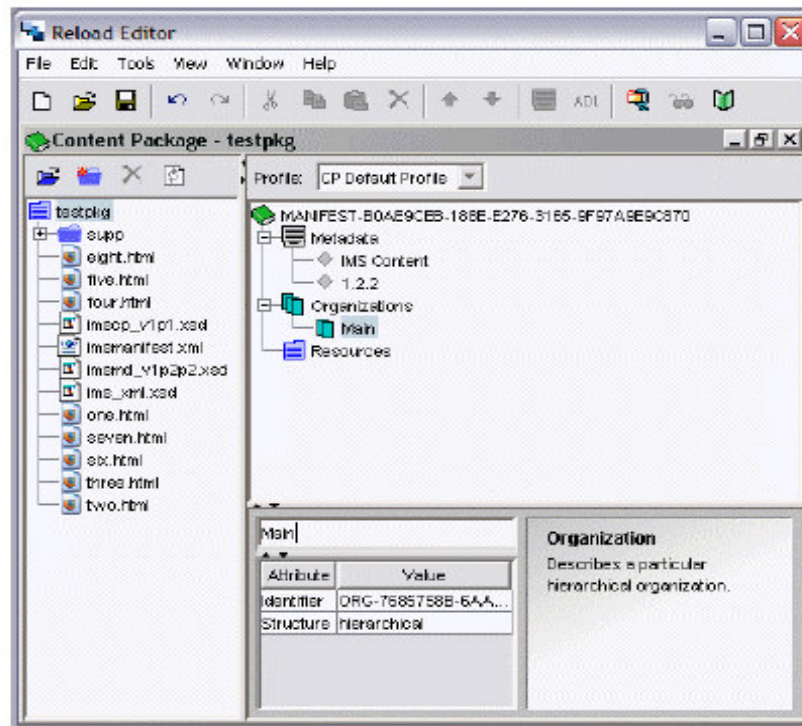


Figura 11: Herramienta RELOAD (parte 4)

La siguiente etapa, necesaria como la que más, es añadir ítems al paquete de contenidos. El procedimiento es similar, por lo que no se va a hacer un excesivo detalle en el mismo, simplemente se puede pulsar en Edit-Add ítems y seguir los pasos que se van pidiendo, o una función algo más útil, como el “arrastrar y soltar”, que permite que el recurso seleccionado en el panel de recursos se pueda arrastrar al panel de ítems y soltar consiguiendo el objetivo de crear un ítem en el paquete que se está editando. Una vez que se han añadido todos los ítems, la pantalla principal muestra los tres paneles de la forma que sigue:

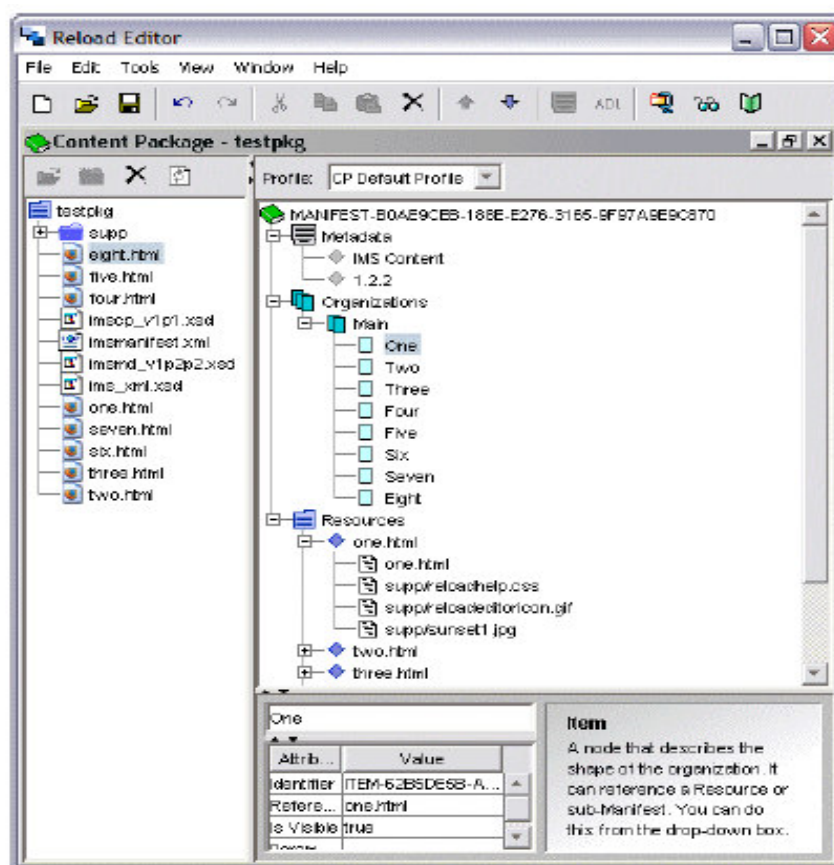


Figura 12: Herramienta RELOAD (parte 5)

Con el paquete de contenidos ya creado, el RELOAD nos ofrece la posibilidad de visualizar el mismo, en forma de explorador, de tal forma que se pueden ir seleccionando los recursos o las organizaciones creadas, de manera que en la pantalla serán mostradas las mismas, con un aspecto como el que se muestra a continuación de este texto:

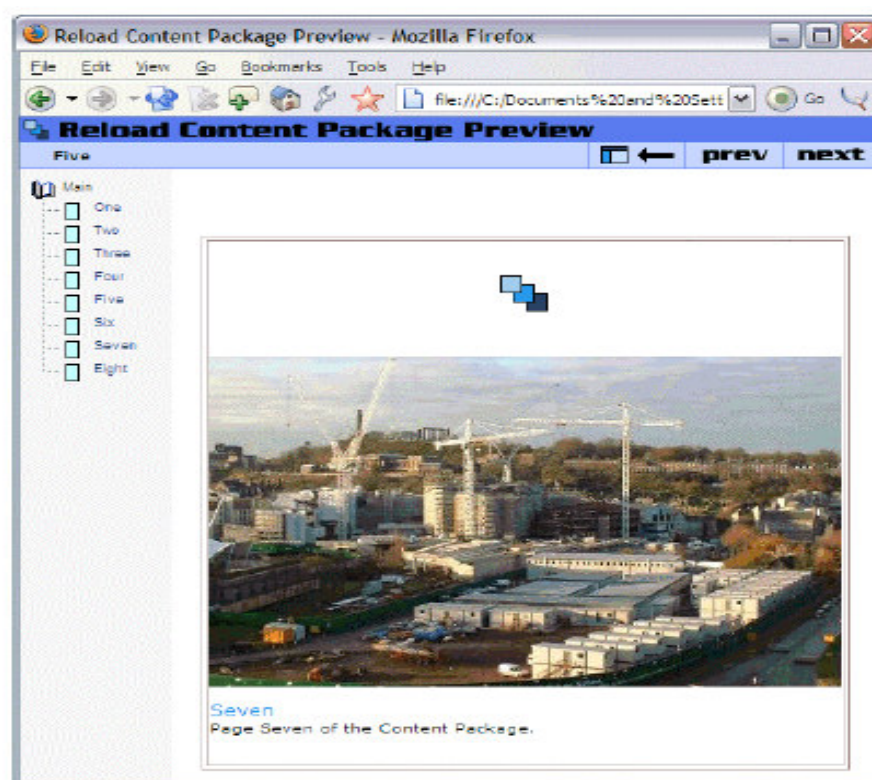


Figura 13: Herramienta RELOAD (parte 6)

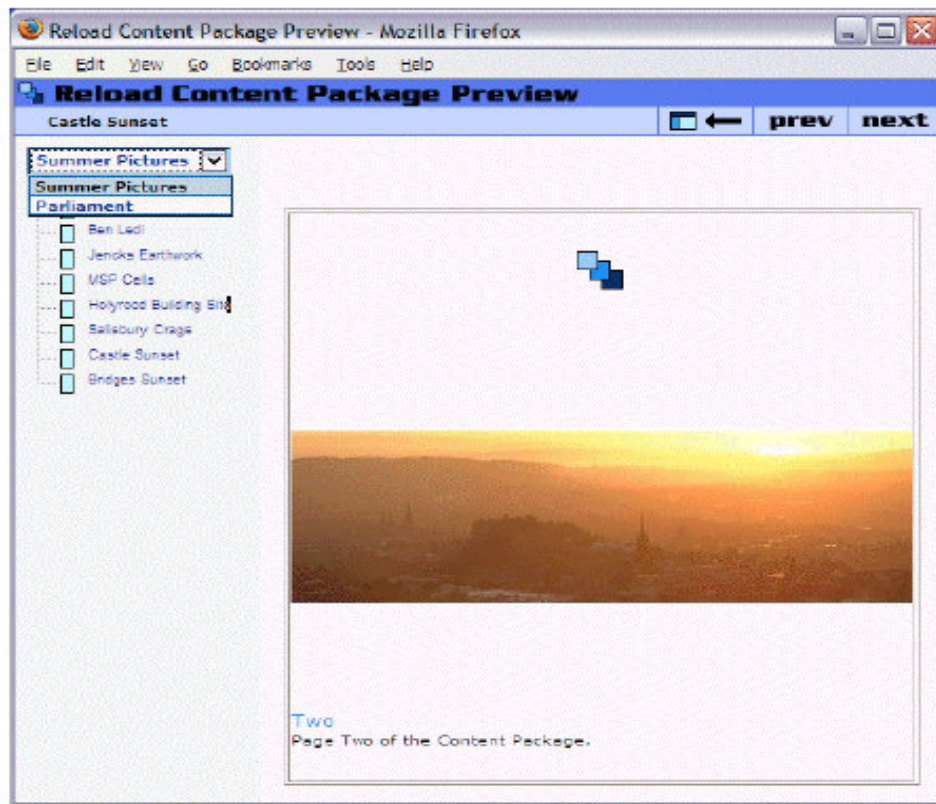


Figura 14: Herramienta RELOAD (parte 7)

El RELOAD nos ofrece la opción de visualizar una vista del paquete de contenidos en forma de HTML o en forma de archivo ZIP con la estructura que sigue el estándar de paquete de contenidos IMS Content Package, el seleccionar una de las dos opciones no tiene más historia que acceder al menú contextual File y elegir la que más nos interese, o ambas por separado si eso es lo que queremos.

Finalmente, el RELOAD nos ofrece la posibilidad de editar manualmente el `imsmanifest.xml`, es decir, modificar las distintas partes del mismo, pero para esto hay que tener un conocimiento a nivel de experto de XML y del estándar IMS Content Package, dado que una modificación no además puede hacer que se pierda toda la coherencia de la unidad creada, con lo que no podríamos obtener un paquete de contenidos utilizable en otros ámbitos de E-learning que nos interese.

En resumen:

Ventajas de RELOAD

- Es un editor gráfico con una interfaz bastante amigable, en la que se observan en todo momento todas las partes del paquete de contenidos que está siendo creado.
- Permite añadir y eliminar recursos, organizaciones y metadatos con clicks de ratón.
- Permite el arrastrar y soltar para crear item a partir de recursos ya añadidos al paquete de contenidos.
- Permite visualizar el paquete de contenidos creado.
- Es de código abierto.

Inconvenientes de RELOAD

- Es necesario diseñar detalladamente el fichero XML del manifiesto que forma parte de la unidad de aprendizaje, por lo que es necesario un conocimiento de nivel muy alto de XML para añadir todos los campos que son obligatorios en los paquetes de contenido, lo que dificulta mucho su uso y hace que no pueda ser utilizado por cualquier educador.
- El idioma en que está el editor es único, es el inglés, lo que no es un inconveniente importante, pero si puede dificultar en algunos momentos que el usuario entienda algunos tecnicismos.

Por lo que hemos visto, el RELOAD es un buen editor, con muchas funcionalidades para la creación y edición de paquetes de contenidos que sigan el estándar IMS Content Package, teniendo pocos inconvenientes y muchas ventajas, pero el inconveniente principal que tiene hace que su utilidad baje mucho, puesto que aparte de existir pocos educadores familiarizados con el XML, estos suelen ser muy reacios a tener que aprender un formato nuevo para realizar sus cursos, lo que provoca una reacción negativa ante la herramienta y una tendencia a aferrarse al sistema tradicional de enseñanza. Uno de los objetivos de este Proyecto es el de eliminar esta barrera habiendo que el usuario que quiere crear un paquete de contenidos no tenga que conocer absolutamente nada de XML, con lo que se elimina la principal barrera existente en esta herramienta y se daría un paso de gigante en el campo de las aplicaciones de autoría de materiales educativos.

3.1.1.2. MD2: Método de desarrollo de materiales didácticos

El análisis de las soluciones actuales a la problemática del desarrollo de los materiales didácticos como RELOAD llevó a la autora de la solución MD2: Método de desarrollo de materiales didácticos [13] a plantear que: El desarrollo puede ser soportado de forma efectiva si se cuentan con los medios que permitan describir sus componentes fundamentales, es decir, los contenidos y la estrategia pedagógica, así como sus propiedades deseables, tales como el carácter reutilizable, la calidad, la usabilidad, la utilidad o valor pedagógico y el cumplimiento con los estándares y especificaciones de e-learning. Además es preciso contar con un método que guíe las diferentes etapas del desarrollo: selección, composición y evaluación.

El marco conceptual propuesto por la solución está compuesto por un modelo, un método y la arquitectura para la herramienta de autoría.

El modelo permite describir de manera general los componentes del material: contenidos y estrategia pedagógica; los requisitos sobre sus características deseables a través de elementos que se han agrupado en cuatro vistas: Dominio de Conocimiento (DC), Pedagógica (P), Soporte (S) y Calidad-Usabilidad (C-U). Estos elementos proporcionan información de carácter organizativo, pedagógico y tecnológico para que sea posible obtener un diseño armónico de un proceso educativo basado en e-Learning. En el meta-modelo se definen conjuntos de relaciones entre los elementos que incluyen correspondencias entre las descripciones de alto nivel técnico provenientes de los estándares e-Learning con descripciones de los requisitos del material más simples y cercanos al lenguaje de sus creadores. Gracias a ellas, se puede facilitar el desarrollo y guiar a los diferentes tipos de desarrolladores durante sus diversas etapas sin que éstos tengan conocimientos sobre los detalles de las especificaciones y estándares e-Learning para obtener materiales conformes a ellos.

El método está compuesto por 5 pasos: Entrada de requisitos, Selección de recursos, Composición, Evaluación y Generación de anotaciones semánticas. Estos pasos utilizan las respuestas de los desarrolladores a un conjunto de preguntas sobre los requerimientos del material que son descritos utilizando un conjunto mínimo de elementos del modelo. Los algoritmos definidos para los 5 pasos del método permiten definir guías y mecanismos pueden ayudar a los desarrolladores en la elección de cuál es el procedimiento que deben seguir durante la etapa de selección de acuerdo con sus necesidades específicas; guías y mecanismos para la agregación e integración de los recursos en la estructura del material durante la etapa de composición; mecanismos que permiten llevar a término la evaluación del material obtenido en relación con su utilidad pedagógica y la usabilidad de su interfaz, además de facilitar las labores de rediseño en aquellos casos en que dichas cualidades no sean las adecuadas para que el material sirva de soporte efectivo a un determinado proceso educativo. El potencial de reutilización del material se asegura no sólo desde la forma en que se define la composición sino también en el paso de Generación de anotaciones semánticas que permite añadir información de carácter semántico al material creado para permitir su localización, recuperación. Éste además se encarga de la creación del paquete de

distribución del material conforme con la especificación IMS CP, que permitirá la interoperabilidad del material entre sistemas heterogéneos conformes con la especificación.

Modelo y método son las bases para la arquitectura de herramientas de autoría de carácter generativo. Dicha arquitectura está formada por un conjunto de módulos que utilizando la información proporcionada por los elementos del modelo, están encargados de la implementación de los algoritmos definidos para cada uno de los pasos del método que permiten guiar y ofrecer soluciones a cada una de las etapas del desarrollo y aseguran que el material obtenido exhiba las características deseables.

El presente proyecto tiene el objetivo de implementar la solución **MD2: Método de desarrollo de materiales didácticos** para el caso en que los materiales didácticos son actividades de evaluación basadas en la especificación IMS LD.

3.2. Tecnologías utilizadas

3.2.1. XML

XML, sigla en inglés de Extensible Markup Language (“lenguaje de marcas extensible”), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C) [14]. Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

XML proviene de un lenguaje inventado por IBM en los años setenta, llamado GML [15] (Generalized Markup Language), que surgió por la necesidad que tenía la empresa de almacenar grandes cantidades de información. Este lenguaje gustó a la ISO, por lo que en 1986 trabajaron para normalizarlo, creando SGML (Standard Generalized Markup Language), capaz de adaptarse a un gran abanico de problemas. A partir de él se han creado otros sistemas para almacenar información.

El W3C es la institución responsable de la creación, el desarrollo y el mantenimiento de las especificaciones XML. El XML Core Working Group es responsable de varias especificaciones, algunas de las cuales están publicadas como recomendaciones del W3C. XML 1.0 fue publicado originalmente como una recomendación del W3C en febrero de 1998. Las más recientes recomendaciones sobre XML tratan sobre XML 1.0 y XML 1.1.

El objetivo del XML es la estructuración de datos, evitando por completo la ambigüedad. El hecho de seleccionar XML como lenguaje para la representación de información debe ser analizado pormenorizadamente ya que conlleva la utilización de soluciones tecnológicas que influirán en el resto del proyecto. Los aspectos a considerar son:

1. El XML es un metalenguaje de marcado que puede ser usado para definir lenguajes de marcado para dominios y espacios de problemas específicos.
2. XML proporciona una estructura lógica y física para describir los datos.
3. Las instancias de XML pueden ser validadas contra una definición formal de una especificación.
4. XML soporta internacionalización.
5. XML es extensible. Al contrario que otros lenguajes de marcado (como HTML), se puede definir nuevas etiquetas (y por tanto nuevos elementos del protocolo) sin ser precisos cambios en el propio XML.

6. XML es texto, por lo que los fragmentos XML son fácilmente creados, editados y administrados mediante herramientas software comunes.

7. XML está evolucionando. Las especificaciones formales están siendo influenciadas y actualizadas a medida que se va adquiriendo más experiencia de uso.

8. Los datos binarios deben ser codificados en forma de texto para poder ser representados en XML.

9. XML es verboso si se compara con otros lenguajes de representación de datos estructurados.

10. Las implementaciones relacionadas con XML son relativamente nuevas. Mientras que los diseñadores e implementadores adquieren experiencia, no es extraño encontrar defectos en los productos.

11. Existe un gran soporte para XML en utilidades de desarrollo software, tanto opensource como propietario.

12. La velocidad de procesamiento de XML puede ser un factor negativo en algunos entornos. El procesamiento XML puede ser más lento debido a que los flujos de datos son más grandes que en otras representaciones y el uso de analizadores genéricos incorpora una capa software adicional, con el asociado coste extra (este coste puede reducirse si se emplean analizadores particulares optimizados).

13. XML es independiente de la plataforma y cumple completamente con el estándar Unicode.

Las principales ventajas del XML son:

- Es extensible (una vez que un xml fue diseñado y puesto en producción, es posible extenderlo con la adición de nuevas etiquetas de modo que los antiguos consumidores puedan continuar utilizando el servicio sin complicación alguna).
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan *bugs* y se acelera el desarrollo de la aplicación.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

XML se encuentra en un estado de madurez muy sólido y su aceptación en el mundo de la Computación ha sido excelente, posiblemente gracias al aval proporcionado por instituciones como el W3C. Existe un gran soporte de lenguaje, no sólo reflejado en el elevado número de herramientas sino también en la cantidad de bibliotecas para lenguajes de programación capaces de procesar este lenguaje.

3.2.2. JDOM

El lenguaje XML, en la actualidad, se ha impuesto como un estándar debido a su potencia para el intercambio de datos entre sistemas, como ya hemos comentado anteriormente. En e-learning, varias especificaciones recomiendan su uso para la descripción de unidades de aprendizaje. Este proyecto ha seguido las especificaciones de IMS LD, todas dispuestas en forma de XML.

Existen varias librerías de código libre que permiten el tratamiento y manipulación de ficheros XML de una forma sencilla, intuitiva y potente. Para comprender el origen de JDom [17], es importante introducir el concepto de DOM (Document Object Model):

- **DOM:** es una interfaz independiente del lenguaje y la plataforma que permite a los componentes software acceder y actualizar directamente el contenido, la estructura y el estilo de documentos HTML y XML. El responsable del DOM es el W3C. Por extensión, DOM también se puede utilizar para manipular documentos XHTML y HTML. Técnicamente, DOM es una API de funciones que se pueden utilizar para manipular las páginas XHTML de forma rápida y eficiente.

- **SAX:** El *Simple API for XML* (SAX) es una interfaz simple para aplicaciones XML. Fácil e intuitiva, muchos programadores la utilizan, ya que se usa especialmente en situaciones en los que los archivos XML ya están en una forma que es estructuralmente similar a la que deseamos obtener. [18]

Entre las librerías especialmente diseñadas para JAVA, la seleccionada ha sido JDOM, que cuenta con numerosas ventajas que se detallan a continuación.

JDOM es un API para leer, crear y manipular documentos XML de una manera sencilla y bastante intuitiva, sobre todo para los programadores de lenguaje JAVA, algo en lo que se separa de DOM y SAX (Simple Api for XML), que están diseñadas independientes del lenguaje, lo que provoca que su utilización resulte bastante más complicada para el programador en cuestión.

La diferencia principal de DOM y SAX con JDOM es que las primeras no dan la opción de trabajar directamente con ellas, para que podamos utilizarlas hemos de hacer uso de los parsers de las mismas, como son: Xerces, XML4j, Crimson,...

La API JDOM no es un parser, usa un parser para su trabajo, JDOM lo que propone es una capa de abstracción en el tratado de documentos XML que hace que el programador se olvide de cómo se realizan las operaciones más próximas al fichero XML, ya que de eso se encarga la API a través de las operaciones que ofrece y que son invocadas por el programador en función de su objetivo.

En el proyecto se ha decidido utilizar JDOM debido a que el lenguaje de programación es en lenguaje Java, y la forma en que JDOM ofrece los servicios de manipulación de XML es la más cercana a Java, dado que no hay que preocuparse de nada más que solicitar una operación y la API se encarga de realizarla. Debido a que la experiencia de quien realiza el proyecto es fundamentalmente Java, el API JDOM es sin duda la más adecuada para la aplicación a desarrollar. Además, dado que los archivos XML se van a generar, y no a leer de estructuras ya creadas únicamente, se considera JDOM más adecuada que SAX, que es más indicado para cuando la estructura del XML es fija e invariante. El parser elegido para la utilización de JDOM es Xerces.

3.2.3. JBoss

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro [19]. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte. Los principales desarrolladores trabajan para una empresa de servicios, JBoss Inc., adquirida por Red Hat en Abril del 2006, fundada por Marc Fleury, el creador de la primera versión de JBoss. El proyecto está apoyado por una red mundial de colaboradores. Los ingresos de la empresa están basados en un modelo de negocio de servicios. JBoss implementa todo el paquete de servicios de J2EE.

JBoss AS es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business. Combinando una arquitectura orientada a servicios revolucionaria con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia. Por este motivo es la plataforma más popular de middleware para desarrolladores, vendedores independientes de software y, también, para grandes empresas.

Las características destacadas de JBoss incluyen:

- Producto de licencia de código abierto sin coste adicional.
- Cumple los estándares.
- Confiable a nivel de empresa
- Incrustable, orientado a arquitectura de servicios.
- Flexibilidad consistente
- Servicios del middleware para cualquier objeto de Java
- Ayuda profesional 24x7 de la fuente
- Soporte completo para JMX

El servidor de aplicaciones JBoss es un servidor que se ha demostrado que rinde bien en aplicaciones con poco tráfico concurrente, y el hecho de ser de código abierto y libre es un factor importante en la elección del mismo. A su vez, dado que el Proyecto se ha desarrollado en Java, se ha considerado conveniente elegir una alternativa desarrollada también en Java, y dado que ya se tenía experiencia previa en la utilización del mismo, pues se ha optado por esta alternativa como el servidor de aplicaciones, pero no existe ningún tipo de problema en desplegarla en otro servidor, como pueda ser Apache Tomcat.

3.2.4. Struts

Struts es un framework que permite desarrollar aplicaciones WEB que implementan el Modelo Vista Controlador en el lenguaje Java [20]. Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Un Framework es un conjunto de clases, interfaces, módulos y librerías que cooperan conjuntamente para solucionar un determinado problema. Struts, siguiendo el concepto de Framework, está formado por un conjunto de clases, servlets y una serie de librerías de etiquetas o taglibs que se utilizan en las páginas JSP, para acceder a objetos del modelo, de forma que estas páginas queden libres de código Java.

Struts ofrece una serie de soluciones para cada una de las partes del patrón MVC, de esta forma el modelo puede estar formado por clases Java y por clases que heredan de la clase Action y ActionForm instanciadas desde el propio Struts, que son extensiones de un servlet. Los datos del modelo son pasados a la vista para ser mostrados al usuario. El papel de controlador lo realiza un servlet que se configura en un fichero XML, en él mismo se asignan las clases del modelo a peticiones del usuario y la vista que se mostrará como respuesta a las mismas. Struts también posee mecanismos que le permiten realizar validaciones de formularios de entrada, validaciones de errores y para internacionalización de la vista.

En el patrón MVC, el servlet controlador está proporcionado por struts, las páginas JSP son la vista y la lógica de negocio de la aplicación reside en clases Java.

En Struts, el servlet controlador dirige las peticiones http de los navegadores Web a otros objetos del Framework, que en su mayoría serán subclases de Action. El controlador analiza el fichero de configuración de que dispone, `struts-config.xml`, para realizar esta tarea. El objeto Action maneja la solicitud del cliente y devuelve un objeto `ActionForward`. El mapeo del mismo se encuentra de nuevo en el fichero de configuración de Struts y dirigen la vista hacia una página JSP. Si nos encontramos en una operación cuyo resultado pueda ser éxito o fracaso, en caso de éxito se redirige a una página que permita continuar con el desarrollo de la actividad en la aplicación, y en caso contrario, se puede redirigir a una página de error.

Uno de los asuntos más importantes en una aplicación WEB es retener y validar los datos de entrada que son introducidos por el usuario. Con Struts, se pueden definir un conjunto de clases que almacenan y validan los formularios de entrada. Los objetos Action van a recibir parámetros con la forma de estas clases denominadas `ActionForm`. Las clases que validan estos formularios van a estar definidas en el fichero de configuración de Struts, y el controlador va a buscar en el mismo a la hora de realizar la ejecución del flujo de la aplicación en cada caso concreto.

3.2.5. Coppercore y SLED

El Framework Coppercore es un proyecto de la Universidad Abierta de Holanda (OUNL) que consiste en un motor de la especificación IMS LD capaz de ejecutar Unidades de aprendizaje de los tres niveles [21]. Está implementado de forma que permita ser integrado con otro tipo de aplicaciones que requieran hacer uso de los servicios de validación, publicación,... de paquetes de contenidos, escondiendo la complejidad de la ejecución para que sea utilizado por los desarrolladores.

Hace uso de XML y Java como base de programación e intercambio de información, ya que IMS LD está basado en XML.

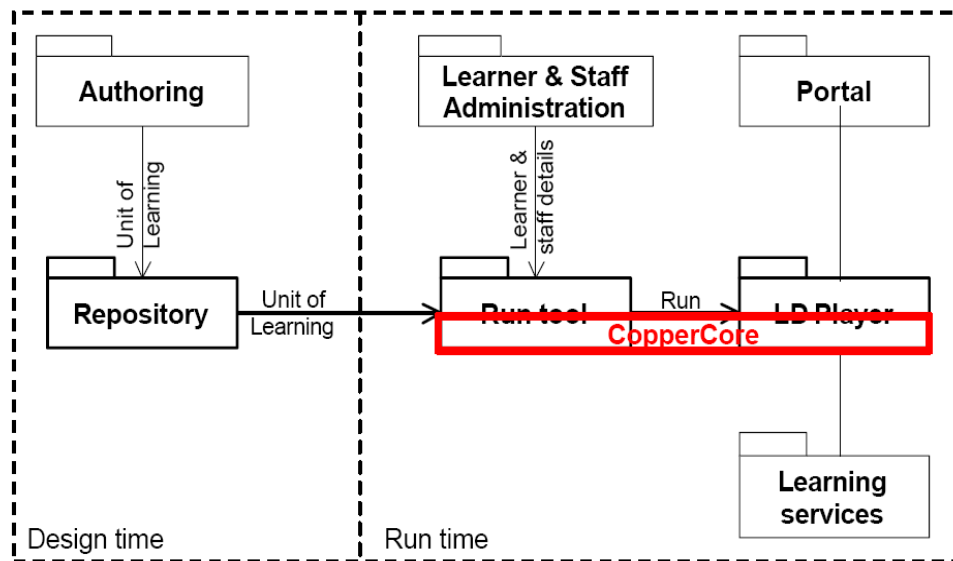


Figura 15: Estructura de la interacción entre SLED y Coppercore

El Framework de Coppercore provee una serie de servicios a los desarrolladores que decidan integrarlo en su aplicación, tales como pueden ser:

- Crear usuarios.
- Crear roles.
- Asociar usuarios a roles.
- Crear ejecuciones.
- Publicar unidades de aprendizaje.
- Visualizar unidades de aprendizaje.

Es muy utilizado en aplicaciones WEB de E-Learning debido a que provee una interfaz que hace que el desarrollador se olvide de como realizar las acciones de publicación, simplemente necesita instanciar un Adaptador de Servicios de Coppercore, y puede a través de este invocar las distintas acciones que permite realizar, permitiendo así que la integración en una aplicación WEB sea relativamente sencilla, al centrarse el desarrollador en la parte WEB y en su lógica particular, sin preocuparse de cómo el Coppercore va a realizar las acciones solicitadas.

Un ejemplo claro de aplicación WEB que utiliza el Coppercore es SLED, una aplicación WEB desarrollada en Struts que permite realizar las acciones que ofrece Coppercore a través de una interfaz sencilla, en la que se debe realizar previamente un Login, para poder acceder a las distintas unidades de aprendizaje publicadas [22].

El código de SLED está estructurado de la siguiente forma:

- Paquete coppercore: contiene todas las clases necesarias para hacer uso de las funcionalidades implementadas por Coppercore, en estas clases se realiza la instanciación de los adaptadores y se permite realizar la invocación de los servicios que necesite el SLED en cada caso.

- Paquete sled: este paquete está dividido en subpaquetes, de los que los más importantes son:

- admin: este paquete contiene todas las funcionalidades que permite realizar el SLED, en forma de Action de Struts, de forma que en el código de las mismas se realizan las invocaciones a los servicios de Coppercore para poder completar las tareas solicitadas.

- login: este paquete contiene toda la funcionalidad necesaria para el control de las operaciones de login y su correspondiente vista de la aplicación en función de los usuarios, es decir, que cada usuario visualice sólo aquello que el ha creado o a lo que le han dado acceso.

- player: este paquete contiene todas las acciones del visualizador de unidades de aprendizaje. Es el que se va a encargar de realizar toda la lógica necesaria para poder ver los paquetes de forma estructurada y poder interactuar con ellos.

- system: este paquete contiene toda la lógica de negocio necesaria para poder controlar el flujo de control de la aplicación, así como las clases propias de la aplicación como pueden ser excepciones o configuraciones de conexión con el Coppercore.

En el proyecto desarrollado, se ha integrado la funcionalidad de Coppercore haciendo uso de la aplicación Web SLED, modificando el código que se ha considerado oportuno para adaptarlo al entorno de autoría de materiales didácticos desarrollado, permitiendo así la visualización de los materiales didácticos que se crean a través de la aplicación WEB desarrollada. Esto es una medida que se modificará para la posterior integración con el motor de visualización que se adopte como definitivo en la aplicación WEB global.

4. Análisis del sistema

4.1. Finalidad del sistema

Con el desarrollo de las nuevas tecnologías, en todos los campos de la vida cotidiana se han ido introduciendo herramientas relacionados con el ámbito informático cuya finalidad es facilitar y agilizar las diversas actividades de la vida diaria de cualquier trabajo. La enseñanza no iba a ser menos, y poco a poco se han ido introduciendo herramientas que permitan aprovechar el poder de la tecnología con el fin de que los alumnos aprenden de forma más rápida y cómoda, y a su vez los profesores dispongan de mejores medios para poder realizar su labor.

Esto se ha visto reflejado como ya hemos comentado en la aparición de los distintos tipos de paquetes de contenidos diseñados para su uso en entornos informáticos, así como algunas herramientas están diseñadas para asegurar el cumplimiento de los estándares y especificaciones sobre e-Learning y se dirigen especialmente a un sector pequeño de los profesionales involucrados en la creación de los materiales, aquellos que tienen un conocimiento profundo de estos estándares. En estas herramientas no se ha considerado resolver las interrogantes que surgen a lo largo del desarrollo tales como dónde localizar los contenidos más apropiados para el material, cómo recuperar dichos contenidos, cuáles son los criterios más indicados para seleccionar los contenidos, cómo integrar estos contenidos en el material, cuándo y cómo controlar la coherencia, completitud y precisión de los contenidos seleccionados o cómo asegurar el carácter reutilizable del material tomando en cuenta la naturaleza reutilizable de sus componentes. Otras herramientas, las basadas en modelos, proporcionan soluciones parciales al desarrollo, toman en consideración sólo uno de los componentes fundamentales de los materiales: contenidos o estrategias pedagógicas y por lo general proporcionan mecanismos para resolver los problemas de la etapa de composición pero no solucionan las cuestiones relacionadas con las etapas de selección y evaluación.

Como consecuencia, la problemática de soporte a la fase de desarrollo de materiales didácticos se puede resumir en los siguientes puntos:

- Necesidad de disponer de descripciones sobre los componentes del material y los requisitos para su desarrollo, en los que se incluyan las propiedades deseables como reutilización, calidad, valor pedagógico y el cumplimiento de los estándares y especificaciones.
- Ausencia de métodos de desarrollo que incluyan:
 - Guías y mecanismos que ayuden a los desarrolladores (expertos o novatos) en la elección de cuál es el procedimiento que deben seguir durante la etapa de selección de acuerdo con sus necesidades específicas.
 - Guías y mecanismos que permitan la agregación e integración de los recursos en la estructura del material durante la composición del mismo.
 - Métricas y reglas que permitan controlar durante la evaluación las propiedades del material obtenido.
- Necesidad de extender la actual audiencia de los usuarios de las herramientas de autoría: desde los profesionales con conocimientos avanzados de los estándares hasta cualquiera que requiera desarrollar materiales didácticos ya sean profesores, tutores, diseñadores instructivos, etc.

Por lo tanto, es necesario contar con un marco de desarrollo que proporcione soluciones sistemáticas y racionales a esta fase de la creación de los materiales y que cubra las actuales carencias. Además es imprescindible contar con entornos de autoría de calidad basados en dicho marco de desarrollo con el objetivo de guiar y apoyar de manera eficaz a los desarrolladores a encarar la complejidad del desarrollo.

4.2. Requisitos de usuario

Una de las primeras fases que se realizó en el proyecto fue la obtención de los requisitos de usuario, que consiste en detallar todos aquellos aspectos de la aplicación que ésta debe cumplir para satisfacer las necesidades del usuario. Para esto, se mantuvieron reuniones a menudo con la tutora del proyecto, a la cuál se le realizaron todas las preguntas necesarias para tener claro todos los aspectos que debía cubrir la aplicación. Estos requisitos se fueron refinando con el paso del tiempo según se iba desarrollando la aplicación, lo que provocó modificaciones en su descripción siguiendo un modelo de desarrollo en cascada que ha afectado a todo el desarrollo del proyecto para llegar a su correcto cumplimiento.

A continuación se detalla de forma exhaustiva cada uno de los requisitos planteados por el usuario, mediante:

- a) Un identificador.
- b) Una descripción del requisito.
- c) Una clasificación de la necesidad de su desarrollo.- Si ese requisito es fundamental o no para el cliente).
- d) Una clasificación de su prioridad.- Cuál es la importancia del requisito en el desarrollo del software.

Requisito Número 1

Identificador: URD001

Descripción: La Aplicación debe permitir al usuario visualizar los diferentes elementos a partir de los cuáles podrá seleccionar aquellos que van a formar parte del paquete de contenidos. En cada uno de los contenidos se visualizará su concepto y su competencia asociada, así como su nombre.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 2

Identificador: URD003

Descripción: La Aplicación debe permitir al usuario seleccionar el tipo de unidad que se va a crear, de entre los tres tipos siguientes: Unidad de aprendizaje basado en problemas, Unidad de aprendizaje basado en procedimientos y Unidad de aprendizaje colaborativo.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 3

Identificador: URD003

Descripción: La Aplicación debe mostrar al usuario un aviso de “imposible realizar una composición” en caso de no haber seleccionado ningún recurso.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 4

Identificador: URD004

Descripción: La Aplicación debe mostrar al usuario una pantalla que marque un error en caso de que no se pueda realizar una composición por problemas ajenos al usuario.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 5

Identificador: URD005

Descripción: La Aplicación debe realizar automáticamente la composición de un paquete de contenidos del tipo seleccionado por el usuario en cada caso que siga los estándares de la especificación IMS LD.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 6

Identificador: URD006

Descripción: La Aplicación debe permitir la publicación del paquete de contenidos creado en un motor de publicación basado en Coppercore.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 7

Identificador: URD007

Descripción: La Aplicación debe permitir al usuario la visualización del paquete de contenidos previamente publicado haciendo uso de un motor de publicación basado en CopperCore.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 8

Identificador: URD008

Descripción: La Aplicación debe estar implementada en lenguaje Java, siguiendo el patrón Modelo Vista Controlador para el diseño y codificación de la misma.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 9

Identificador: URD009

Descripción: La Aplicación debe poder ejecutarse en un servidor de aplicaciones Web, en este caso, el Jboss.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 10

Identificador: URD010

Descripción: La Aplicación debe poder leer y escribir archivos XML para poder crear el fichero de manifiesto de un paquete de contenidos.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 11

Identificador: URD011

Descripción: La Aplicación debe poder crear archivos .zip con los elementos que forman parte de un paquete de contenidos seleccionados por el usuario.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 12

Identificador: URD012

Descripción: La Aplicación debe estar implementada haciendo uso de las tecnologías JSP para las vistas de la misma y Servlets para el control de las peticiones de la misma.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 13

Identificador: URD013

Descripción: La Aplicación debe hacer uso de la tecnología Javascript para realizar las validaciones pertinentes sobre los datos de entrada de la misma.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 14

Identificador: URD014

Descripción: La Aplicación debe visualizarse y poder ejecutarse correctamente en los navegadores Internet Explorer y Mozilla FireFox.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 15

Identificador: URD015

Descripción: La Aplicación recibirá como datos de entrada un conjunto de recursos descritos mediante su nombre, concepto, competencia, tiempo de lectura y ruta relativa del recurso.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 16

Identificador: URD016

Descripción: La Aplicación debe realizar la composición del paquete de contenidos siguiendo las premisas especificadas en el punto **5.3. Composición de Materiales Didácticos.**

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 17

Identificador: URD017

Descripción: La Aplicación no debe ser visualmente inadecuada, es decir, debe utilizar unos colores que garanticen su usabilidad y correcta visualización.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 18

Identificador: URD018

Descripción: La Aplicación debe mostrar una ventana en la que se muestre una barra de progreso mientras se realiza la composición del paquete de contenidos concreto.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 19

Identificador: URD019

Descripción: La Aplicación debe estar diseñada de forma que su código provoque una cohesión máxima y un acoplamiento mínimo entre sus componentes.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 20

Identificador: URD020

Descripción: La Aplicación debe estar contenida en un archivo WAR (Web Archive) que sea desplegable en un servidor de aplicaciones Web.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 21

Identificador: URD021

Descripción: La documentación de la aplicación debe generarse en un archivo .doc que contenga toda la información relativa al diseño e implementación de la aplicación.

Necesidad: Es fundamental que la aplicación cumpla este requisito.

Requisito Número 22
Identificador: URD022
Descripción: La Aplicación debe permitir visualizar al lado de cada una de las plantillas de unidades de aprendizaje seleccionables una pequeña descripción de las mismas.
Necesidad: Es fundamental que la aplicación cumpla este requisito.

4.3. Composición de Materiales Didácticos

En el momento de realizar la composición, los distintos recursos que van a formar parte del paquete de contenidos han de ordenarse para pasar a formar parte de las distintas actividades de la unidad de aprendizaje. Para ello, han de seguir un orden que viene marcado por la situación de los distintos tipos de aprendizaje, así como de las competencias que afectan y el verbo que lleva asociada, teniendo en cuenta el tema que va a tratar el recurso. El orden de las distintas competencias viene reflejado en la siguiente tabla, de acuerdo con la solución propuesta por **MD2: Método de desarrollo de materiales didácticos** [26]:

Tipo de aprendizaje	Nivel de aprendizaje- Tipo Competencia	Verbos más utilizados
<i>Cognitivo</i>	<i>Conocimiento</i>	Conocer-Reconocer-Saber-Definir-Memorizar-Repetir- Listar-Recordar-Nombrar-Relacionar (comolista)-Citar-Exponer-Plantear- Decir-Identificar-Percibir-Catalogar-Clasificar-Etiquetar
	<i>Comprensión</i>	Comprender-Replantear-Discutir-Describir-Reconocer-Tratar-Explicar-Expresar-Identificar Localizar- Informar-Revisar- Contar-Convertir-Interpretar-Exponer-Responder-Entender
	<i>Aplicación</i>	Aplicar-Traducir-Interpretar-Utilizar-Demostrar-Representar-Practicar-Ilustrar-Operar-Programar Esbozar-Relacionar(asociar)-Preparar-Mostrar-Iniciar-Influir-Expresar-Desempeñar-Cumplir
	<i>Análisis</i>	Analizar-Distinguir-Diferenciar-Calcular-Experimentar-Probar-Comparar-Contrastar-Criticar Diagramar-Inspeccionar- Debatir-Inventariar-Preguntar-Relacionar Asociar-Discriminar-Bosquejar-Ajustar-Adaptar-Clasificar
	<i>Síntesis</i>	Sintetizar-Componer-Planificar-Proponer-Diseñar-Formular-Ordenar-

		Unir-Reunir Recopilar- Recabar-Construir-Crear- Organizar-Gestionar Preparar-Combinar- Compilar-Desarrollar- Integrar-Modificar
	<i>Evaluación</i>	Evaluar-Juzgar-Valorar- Tasar-Medir-Comprobar- Revisar-Puntuar-Medir Comparar-Puntuar- Seleccionar-Escoger-Pesar- Sopesar-Concluir
	<i>Pensamiento organizado (High order thinking) Aplicar análisis, síntesis y evaluación para resolver problemas complejos</i>	Evaluar-Crear-Conducir- Analizar
<i>Social</i>	<i>Habilidades motoras</i>	Demostrar-Ejecutar- Mover-Mostrar
	<i>Valores y Actitudes</i>	Respetar-Demostrar- Expresar
	<i>Comportamiento social</i>	Representar (perform)

Una vez que se han seleccionado los recursos que formarán el paquete de contenidos, se incluirá una referencia a cada uno de ellos en el item resources del fichero XML del manifiesto. Este elemento tiene la siguiente forma:

A continuación, se han de asociar cada uno de los recursos introducidos a un environment (entorno) determinado, haciendo que de esta forma se tengan todos los environments (entorno) en el fichero XML de la siguiente forma:

Una vez hecho esto, se asocia cada environment que trate un concepto y una competencia determinada a una actividad. En las actividades, se tendrá en cuenta el recurso, ya que cada actividad puede tener un tema de teoría y unas lecturas de ese tema. Para hacer esta diferenciación, se tendrá en cuenta el tipo de archivo que se ha seleccionado en el recurso, siendo un tema de teoría un archivo .zip, y una lectura un archivo html. La forma de una actividad en un fichero XML es como sigue:

En la creación de estructuras de actividades, es necesario tener en cuenta el tema que trata cada actividad, se realizará la agrupación de las actividades que tengan el mismo tema en una estructura de actividades. La definición de una estructura de actividades en el fichero de manifiesto queda de la forma tal que así:

En cada una de las unidades de aprendizaje, se crea una ejecución, que se ve de la siguiente forma en el fichero XML de descripción del paquete de contenidos en cada caso:

5. Herramienta de composición de materiales didácticos

5.1. Arquitectura

La arquitectura de la aplicación desarrollada sigue el patrón Modelo-Vista-Controlador [23], ya que uno de los principales problemas a la hora de desarrollar aplicaciones, es el posterior mantenimiento y la reutilización del código. Puede darse el caso de que haya que realizar mejoras o cambiar totalmente el aspecto externo de la aplicación, algo bastante común en entornos WEB, pero sin variar la lógica de negocio, por lo que es conveniente separar la capa de presentación de la capa de negocio de la misma, y a su vez de los datos. Esto no es posible con el modelo básico de programación estructurada, ya que en dicho escenario, cualquier cambio que haya que realizarse en el código implica automáticamente realizar cambios que afectan a todo el código, con lo que ello supone de tiempo y esfuerzo para el programador, por no hablar en el caso muy habitual de que el que deba realizar los cambios no sea el que ha realizado el código originario, lo que incrementa exponencialmente el coste de la operación.

El patrón Modelo-Vista-Controlador permite que esta labor sea mucho más fácil, ya que diferencia claramente la aplicación en tres partes, por lo que si se respeta estrictamente el patrón, no será necesario modificar todo su código para modificar la parte que se pretenda de la aplicación.

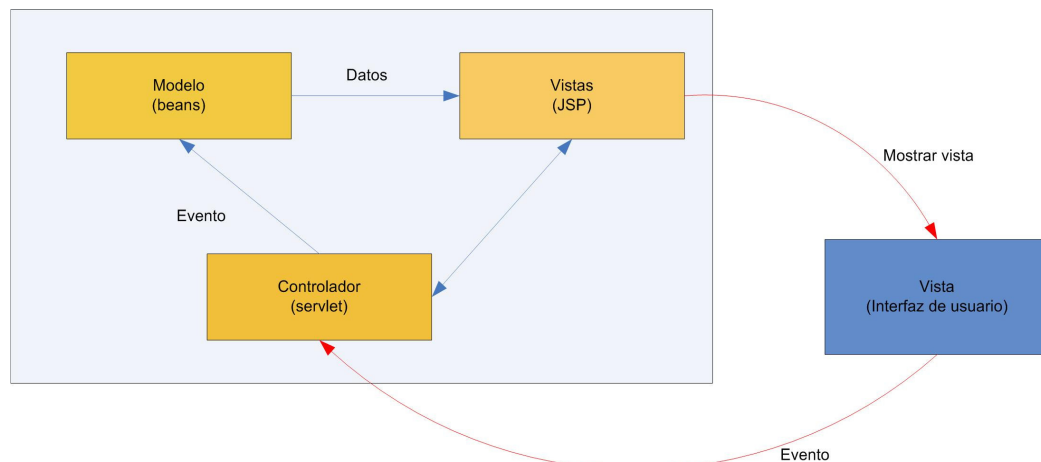


Figura 16: Estructura del Modelo-Vista-Controlador

Las tres capas en la que se estructura la aplicación que sigue esta especificación son:

- **Modelo.** El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.
- **Controlador.** El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.
- **Vista.** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

En este tipo de diseño, el procesamiento de las operaciones se divide entre las tres partes. El controlador recibe un evento o solicitud, que es analizado por el mismo, y basándose en su mapa de configuración, decide que parte o partes del modelo van a ser las encargadas de llevar a cabo la ejecución. El modelo realizará las operaciones que sean necesarias para atender a la petición, accediendo a base de datos u otro tipo de repositorio en caso de necesitarlo. Una vez que termina su ejecución, devuelve el flujo al controlador, que se encargará de presentar al usuario la vista que corresponda en cada momento.

Las principales ventajas de este modelo son:

- Las tres capas del modelo son independientes y se pueden implementar por separado.
- Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su lógica de negocio).
- Permite definir varias vistas para una misma aplicación, por ejemplo, una vista para WEB y otra para una aplicación de escritorio.
- Mayor claridad en el diseño.
- Facilita el mantenimiento. Cada una de las capas puede ser actualizada sin modificar el resto.

Sin embargo, este patrón presenta algunos inconvenientes, que son:

- Mayor complejidad. El patrón MVC introduce nuevos niveles de introducción, incrementa ligeramente la complejidad de la solución.
- Si el modelo cambia rápidamente, puede llegar a sobrecargar a las vistas con las solicitudes de actualización o mantenerlas desactualizadas.

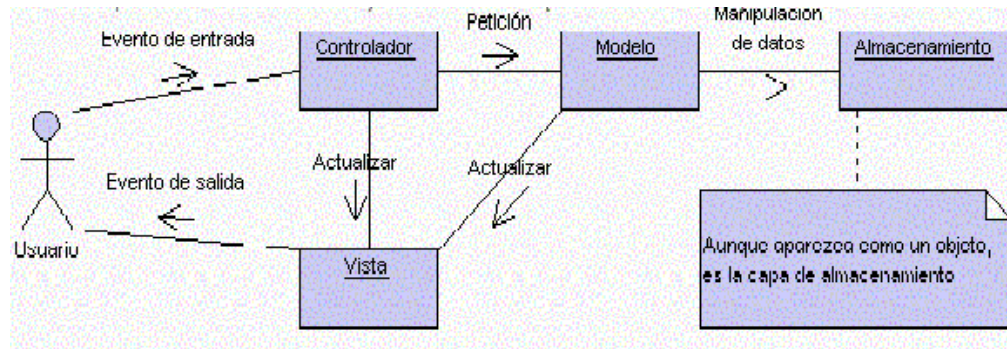


Figura 17: Representación del patrón MVC y su interacción con el usuario.

5.2. Casos de uso

El objetivo de este apartado es analizar a través de los diagramas de casos de uso las distintas funcionalidades que se pueden realizar en la aplicación, haciéndose valer para ello de una herramienta de modelado UML que nos permite realizar estos diagramas de manera que se puedan observar los elementos principales del mismo, que son el actor y el caso de uso que toque en cada apartado, así como las asociaciones entre los elementos que formen parte del caso de uso comentado.

Caso de uso 1: Visualizar Materiales Disponibles

- **Actor:** Usuario de la aplicación.

- **Descripción:** El usuario de la aplicación, una vez que ha realizado una búsqueda de materiales didácticos, puede visualizar en una tabla los diferentes elementos recuperados, teniendo dicha tabla el título del recurso, el concepto que trata, la competencia con que está relacionada, el tiempo de duración asociado y el tipo de recurso del mismo.

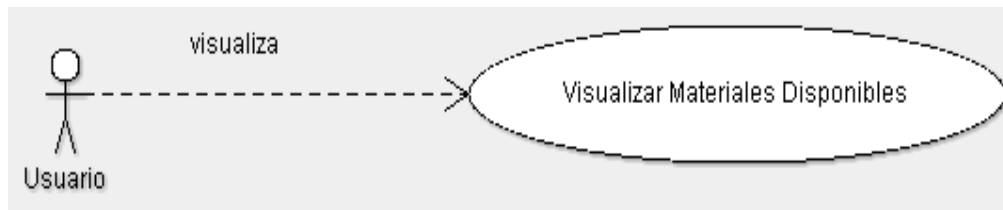


Figura 18: Caso de Uso 1: Visualizar Materiales Disponibles

Caso de uso 2: Examinar Materiales Didácticos

- **Actor:** Usuario de la aplicación.

- **Descripción:** El usuario de la aplicación, una vez que ha realizado una búsqueda de materiales didácticos, puede examinar los diferentes materiales didácticos disponibles en el programa adecuado para cada tipo de elemento, para de esta forma tener una visión clara de todos los componentes que tiene a su disposición para crear un paquete de contenidos que él considere adecuado a sus fines.



Figura 19: Caso de uso 2: Examinar Materiales Didácticos

Caso de uso 3: Seleccionar Materiales Didácticos para Composición y Realizar Composición

- **Actor:** Usuario de la aplicación.

- **Descripción:** El usuario de la aplicación, una vez que ha realizado una búsqueda de materiales didácticos, puede seleccionar los materiales didácticos disponibles que él considere oportuno para que formen parte del paquete de contenidos que va a construir. Para ello, los irá seleccionando de una tabla marcando todos aquellos que quiera siempre que siga unas normas que garanticen la coherencia de la composición. Una vez hecho esto el usuario ha de seleccionar el tipo de paquete de contenidos que va a crear de entre los tres de que dispone (Unidad de Aprendizaje basado en la resolución de Problemas, Unidad de Aprendizaje basado en procedimientos y Unidad de Simple Colaboración).

Si ya ha seleccionado los elementos necesarios correctamente, el usuario puede proceder a realizar la composición pulsando el botón habilitado a tal efecto en la interfaz de la aplicación Web.

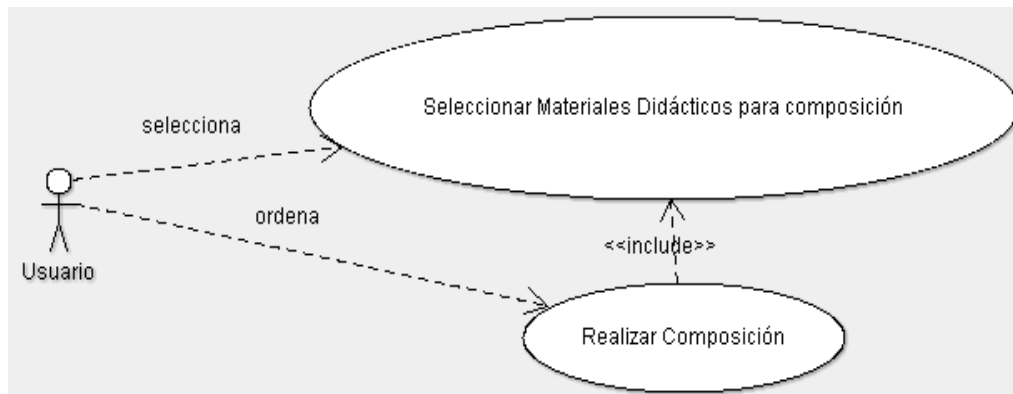


Figura 20: Caso de uso 3: Seleccionar Materiales Didácticos para Composición y Realizar Composición

Caso de uso 4: Publicar Paquetes de Contenidos creados

- **Actor:** Usuario de la aplicación.

- **Descripción:** El usuario de la aplicación, una vez que la aplicación ha realizado la composición del paquete de contenidos en función de sus criterios, puede proceder a realizar la publicación del paquete de contenidos en el motor de visualización basado en Coppercore integrado en la aplicación Web.

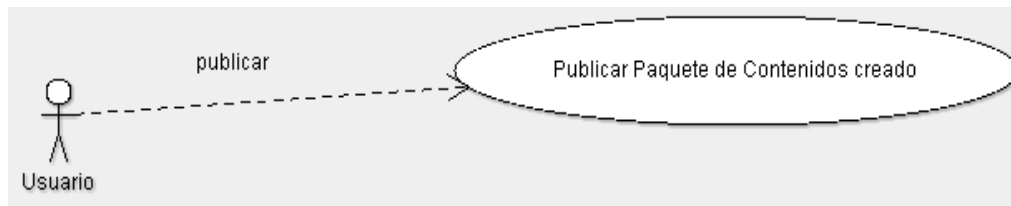


Figura 21: Caso de uso 4: Publicar Paquetes de Contenidos creados

Caso de uso 5: Visualizar Paquete de Contenidos creado

- **Actor:** Usuario de la aplicación.

- **Descripción:** El usuario de la aplicación, una vez que ha realizado la composición del paquete de contenidos previamente creado, puede proceder a descargarlo para posteriormente visualizarlo, ya sea a través del motor de visualización de paquetes de contenido integrado en la aplicación Web, o en su equipo local, para poder comprobar el resultado de la composición que ha realizado.

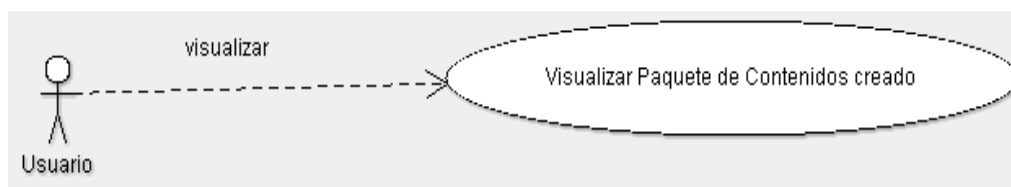


Figura 22: Caso de uso 5: Visualizar Paquete de Contenidos creado

5.3. Diseño detallado de la aplicación

La aplicación, como ya se ha comentado, es una aplicación Web desarrollada en Java, haciendo uso de todas las tecnologías anteriormente detalladas en el apartado 3 de este documento. Este apartado va a permitir visualizar las diferentes clases Java presentes en la aplicación, con su respectiva organización en paquetes y la funcionalidad que desempeña cada una desde un punto de vista técnico.

En cada apartado, se va a hacer referencia también a las decisiones tomadas en cuanto al uso de algunos patrones de diseño, que permite que el código de la aplicación sea más comprensible, reutilizable y modificable en un futuro sin tener que afectar a un elevado número de componentes, permitiendo a su vez que una persona que no lo haya implementado pueda comprenderlo y dejando abierta la posibilidad de modificaciones en un futuro, que debido a que este Proyecto es de un tamaño limitado tanto en funcionalidad como en tiempo no se han afrontado en esta versión, por ello la inclusión de dichos patrones que dejen abierta esa puerta a la futura evolución de esta aplicación Web con la inclusión de nuevas características.

5.3.1. Diseño Gráfico de la Aplicación

En el siguiente apartado se muestra un boceto del diseño que va a seguir la interfaz de la aplicación, ya que éste debe guardar unas pautas para que el usuario no se pierda en la navegación a través de ella, y se mantenga una coherencia visual que permita la usabilidad de la aplicación. El manejo de los estilos queda fuera de este apartado, ya que la implementación realizada, haciendo uso de archivos de “*hojas de estilo en cascada (CSS)*”, permite que el diseño de la interfaz quede totalmente independiente del aspecto final que va a tener la mismas, permitiendo a su vez la realización de manera sencilla y abstracta de una nueva interfaz gráfica, una tarea usualmente destinada a los diseñadores gráficos y no a los programados y analistas del equipo de trabajo implicado en el desarrollo y mantenimiento de una aplicación.

El diseño general de la interfaz gráfica queda de esta forma:

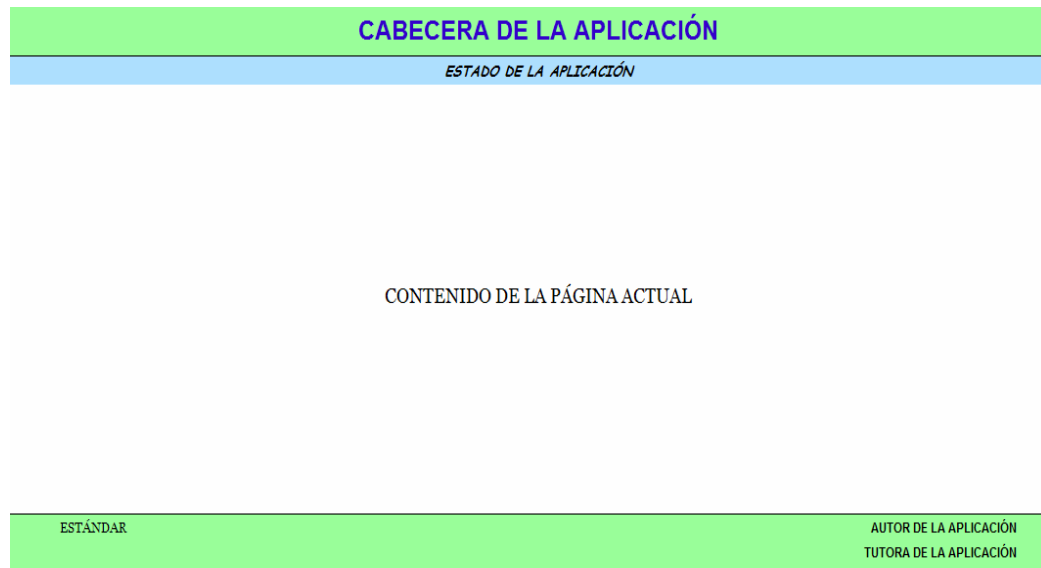


Figura 23: Diseño de la interfaz de usuario de la aplicación

Como se puede observar en el boceto, las partes principales de la interfaz gráfica son:

- *Cabecera de la Aplicación:* En ella se muestra en todo momento el título de la aplicación.
- *Estado de la Aplicación:* En esta zona se muestra un mensaje que le indica al usuario en todo momento en qué fase del proceso de composición se encuentra, lo que le facilita el hecho de no perder la noción de sus acciones.
- *Contenido de la Página Actual:* En esta zona se va a mostrar el grueso de la aplicación, es la parte interactiva de la aplicación, ya que la práctica totalidad de la acción del usuario va a realizarse en dicha ubicación. Esta zona incluirá los distintos botones y formularios de que consta la aplicación según la fase del proceso en que el usuario se encuentre.
- *Estándar:* En este apartado se muestra el estándar que siguen las páginas WEB contenidas en el entorno concreto.
- *Autor de la Aplicación:* Se muestra el autor de la aplicación.
- *Tutora de la Aplicación:* Se muestra en todo momento la tutora que ha dirigido el trabajo.

5.3.2. Diseño técnico de la Aplicación

En el siguiente apartado, se va a proceder a describir los distintos módulos que forman la aplicación “Entorno de autoría de materiales didácticos basados en IMS LD” desde un punto de vista técnico, analizando uno por uno todos ellos y describiendo la funcionalidad de todos. Para ello, por cada módulo se va a mostrar:

- **Nombre del Módulo:** El nombre con el que se identifica el módulo dentro de la aplicación.
- **Descripción general:** Una breve descripción de la funcionalidad que aporta el módulo.
- **Diagrama:** Una descripción visual mediante diagramas UML de los distintos componentes del módulo que se está analizando en el apartado concreto.
- **Funcionalidad específica:** Una descripción de la funcionalidad que aporta el módulo, así como de sus componentes y de detalles de implementación de los mismos, para que se pueda comprender la decisión de diseño adoptada en el contexto de ese módulo.

Este apartado es de vital importancia, ya que va a servir para dar un detalle de las distintas partes que conforman la aplicación, ya que hasta este momento todo se ha analizado desde un punto de vista más funcional y descriptivo que tecnológico, por lo que la información mostrada va a aportar muchos datos sobre lo que hay, pero también sobre la forma de ampliar la aplicación si en un futuro es necesario incluir distintas funcionalidades nuevas o elementos externos en la aplicación de autoría de materiales didácticos.

Es importante destacar que todas las clases implementadas tienen incluida su correspondiente documentación que ha sido generada con la herramienta JavaDoc y se encuentra en el CD que acompaña este documento, por lo que cualquier aclaración específica del código puede ser consultada a través de dicha documentación.

5.3.2.1. Modulo de Unidades de Aprendizaje

- **Nombre:** Modulo de Unidades de Aprendizaje

- **Descripción General:** este modulo representa las unidades de aprendizaje manejadas por la aplicación.

- **Diagrama:**

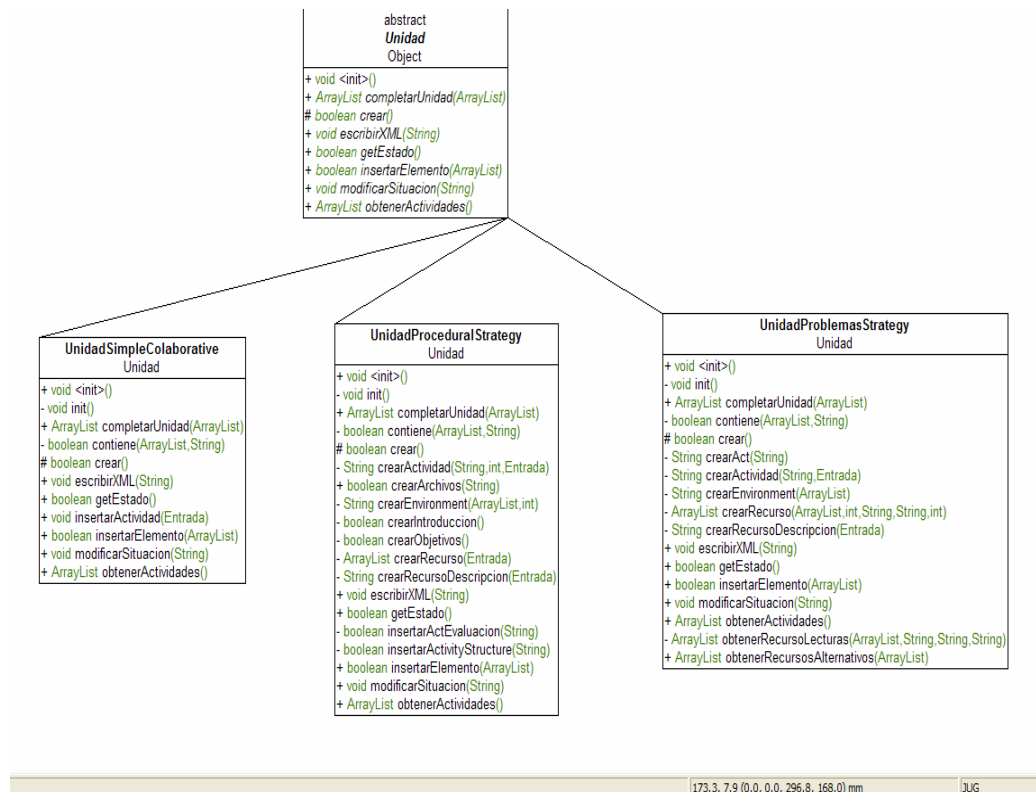


Figura 24: Modulo de Unidades de Aprendizaje

- **Funcionalidad específica:**

El paquete de unidades de aprendizaje se ha implementado de forma que se proporciona una Clase Unidad, la cuál representa las operaciones básicas que van a ofrecer a los objetos que instancien un tipo de unidad, y tres clases que heredan de ella, que son: UnidadSimpleColaborative, UnidadProceduralStrategy y UnidadProblemasStrategy, que son los tres tipos básicos de unidades de aprendizaje manejados en el contexto de la aplicación.

La ventaja de esta implementación es que la inclusión de un nuevo tipo de unidad no representa un impacto grande en la aplicación, ya que como se trabaja con objetos del tipo padre Unidad, si se introduce una nueva, con tal de que implemente todas las operaciones definidas por el tipo Unidad no habría problema.

Las operaciones que se definen en el tipo unidad son operaciones básicas de manejo de la misma, que engloban funciones de creación de la misma, de modificación y comprobación de la integridad, de escritura de la unidad y de acceso a sus elementos fundamentales. Estas operaciones son suficientes para permitir el manejo de las mismas por otros objetos, incluyendo cada tipo de unidad concreto una serie de operaciones que quedan fuera del conocimiento de otras clases, ya que sirven para realizar acciones internas que no es necesario que conozcan otros componentes de la aplicación.

5.3.2.2. Módulo de Fábricas de Unidades

- **Nombre:** Módulo de Fábricas de Unidades.

- **Descripción general:** Este módulo se encarga de gestionar las fábricas que instancian los diferentes tipos de unidades de aprendizaje.

- **Diagrama:**

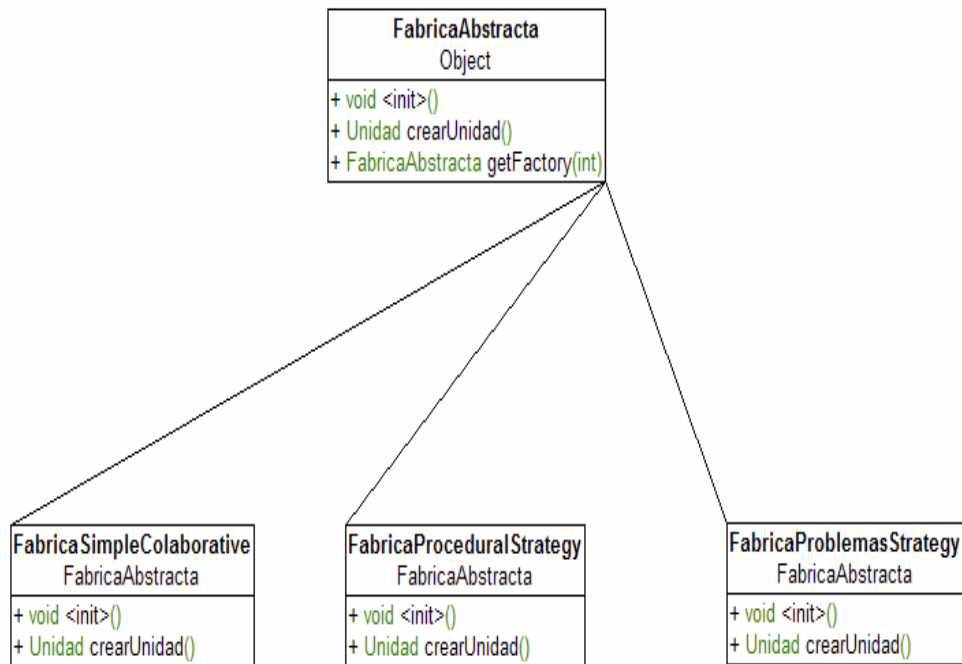


Figura 25: Módulo de Fábricas de Unidades

- **Funcionalidad específica:**

Este módulo va a ser el encargado de llevar el control de la instanciación de todos los objetos de tipo Unidad que se manejan en el contexto de la aplicación, es decir, los diferentes tipos de unidades de aprendizaje existentes.

Para la implementación de este módulo, se ha decidido la utilización del patrón de diseño Factoría Abstracta (Abstract Factory), basado en la noción de que nuestra aplicación trabaja con una serie de unidades (como si fuera una fábrica) que tienen unas determinadas características. Nuestro programa va a hacer uso de dichas unidades realizando una serie de acciones sobre ellos, como son la creación de las mismas, su composición, la consulta de sus características propias...sin importarle quien le está suministrando los productos. Esto se ha implementado con la existencia de una clase abstracta *FabricaAbstracta*, de la que heredan otras 3, una por cada tipo de unidad de aprendizaje, que son las que se van a encargar de instanciar estas unidades.

La gran ventaja de este patrón es que otorga una gran facilidad para la inclusión de nuevos elementos en la lógica de negocio de la aplicación, en nuestro caso la inclusión de nuevas unidades de aprendizaje, una situación que se presume que pueda darse en un futuro, por lo que la implementación realizada se adapta al problema abordado y permite ofrecer una solución práctica y completa.

5.3.2.3. Módulo de Identificadores de elementos

- **Nombre:** Módulo de Identificadores de elementos

- **Descripción general:** Este módulo se encarga de proporcionar la lógica de los identificadores de los distintos elementos de la aplicación.

- **Diagrama:**

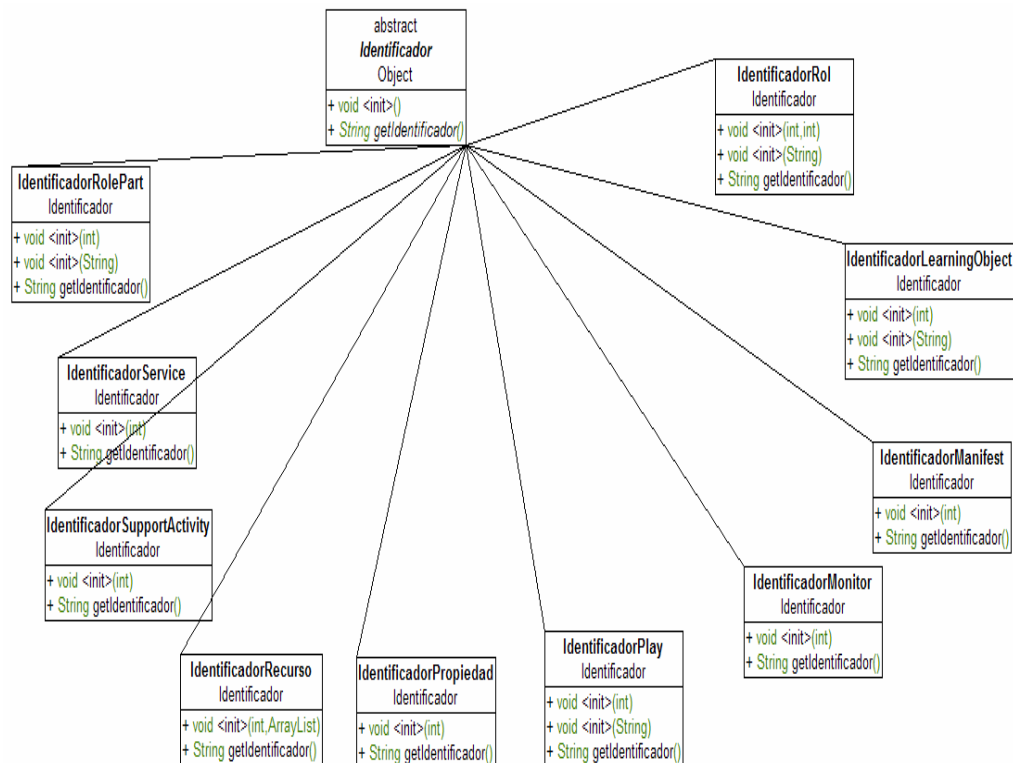


Figura 26: Módulo de Identificadores de elementos

- **Funcionalidad específica:**

Este módulo se encarga de proporcionar los mecanismos de identificación de los distintos elementos que forman parte de la lógica de negocio de la aplicación. En el problema que abraza la aplicación, se da la circunstancia de que hay una enorme cantidad de elementos que necesitan tener un identificador propio, ya que van a formar parte de la unidad de aprendizaje y no puede haber varios elementos que repitan el mismo identificador. Debido a la situación de la estandarización de E-Learning, no existe un código concreto para definir los identificadores, lo que provoca que si en un futuro se impone un estándar, lo más probable es que se imponga un tipo

de identificador determinado para cada elemento, lo que provocaría un gran impacto en el código en otras circunstancias, pero se ha abordado esta cuestión implementado una clase Identificador, de la que van a heredar todas y cada una de las que representan un Identificador para un elemento concreto.

La ventaja de esto es que cualquier cambio futuro de la forma de identificación de los elementos se reduce a la modificación de la clase del identificador concreto, pero dado que la aplicación trabaja con el objeto de la clase que engloba a los identificadores, estos cambios van a ser transparentes a la misma, lo cuál representa una importante ventaja con respecto al mantenimiento evolutivo de la misma.

5.3.2.4. Módulo de Utilidades

- **Nombre:** Módulo de Utilidades

- **Descripción general:** Este Módulo contiene todos los elementos utilizados en la aplicación para realizar acciones que son necesarias para su correcto funcionamiento pero no tienen una existencia suficientemente extensa para representar un módulo por si mismas.

- **Diagrama y descripción general:**

En este módulo se va a comentar la funcionalidad de una forma distinta, ya que por el tamaño de los componentes, se va a ver mucho más claro la definición y la funcionalidad de los mismos si se pone un diagrama y a continuación su descripción y su funcionalidad.

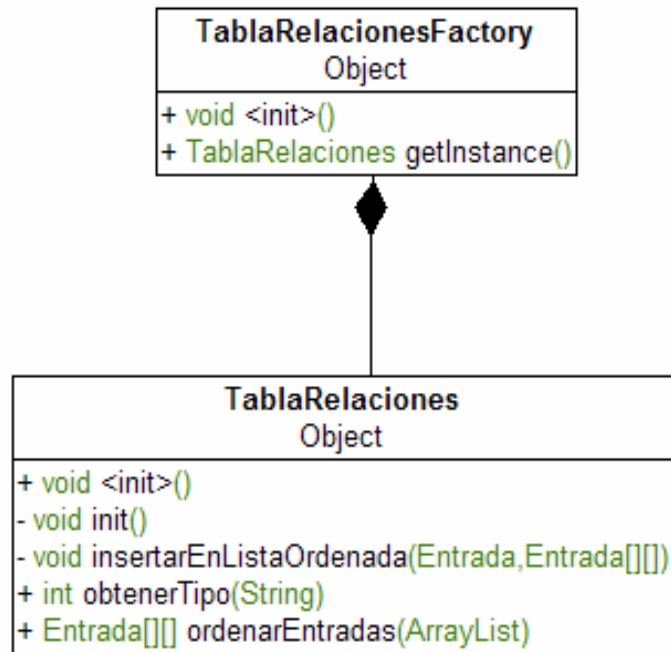


Figura 27: Módulo de Utilidades, Tabla de Relaciones

La Clase **TablaRelaciones** representa una estructura que mantiene ordenados todos los conceptos y competencias que son manejados por la aplicación, de manera que permite manejar un conjunto de Entradas que van a formar parte de un proceso de composición y trabajar con ellas y ordenarlas en base a los criterios definidos por el estándar de E-Learning que se aplica en el contexto de esta aplicación. Es una clase utilizada de manera recurrente en los procesos de composición y creación de unidades de aprendizaje.

El carácter del objeto de la tabla de relaciones hace que vaya a ser utilizado muchas veces pero que su comportamiento no sea dinámico, y dado que sus propiedades no varían con el tiempo, se ha decidido que su acceso se realice a través de la clase **TablaRelacionesFactory**, lo que se traduce en la aplicación de un patrón de diseño Singleton o única instancia, que está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto, que es justo lo que se pretende en la aplicación. La inclusión de este diseño provoca que no se malgaste memoria de la máquina virtual JAVA en la instanciación de objetos innecesaria, provocando una mejora del rendimiento leve en nuestro caso por el tamaño del sistema, pero que en otras condiciones puede ser bastante importante.

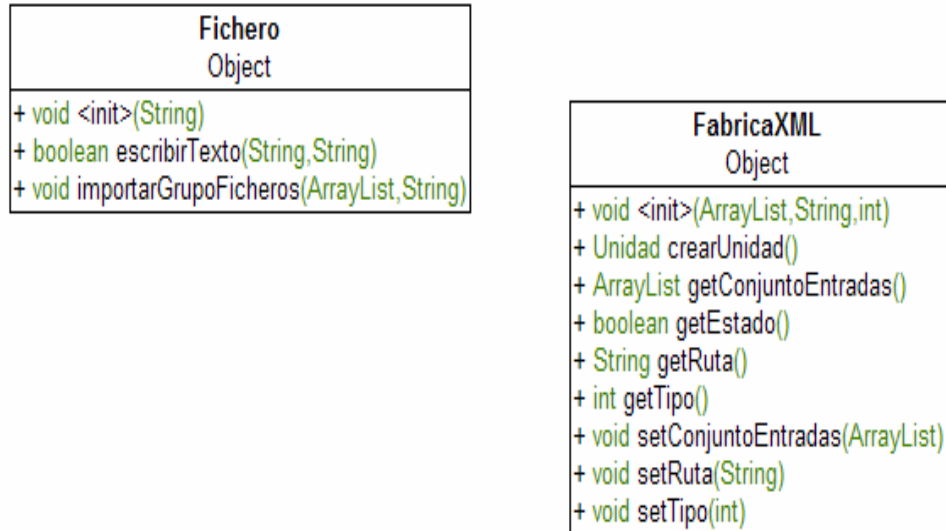


Figura 28: Módulo de Utilidades, Clases alternativas

La clase Fichero se utiliza para realizar todas las operaciones de escritura en ficheros, y de copia de ficheros previamente a la composición de la unidad de aprendizaje en un fichero zip, agrupando las funciones de escritura en disco.

La clase FabricaXML es importante ya que se encarga de construir los ficheros de manifiesto de la unidad de aprendizaje, manejando para ello conjuntos de elementos necesarios para esa labor como las Entradas, siendo la encargada de construir los ficheros físicos que van a formar la unidad de aprendizaje. La diferencia con las otras fábricas es que esta se encarga de trabajar con archivos físicos en vez de elementos lógicos como hace la otra, siendo ésta una clase que se utiliza para acciones muy específicas, no para llevar el negocio de la aplicación.

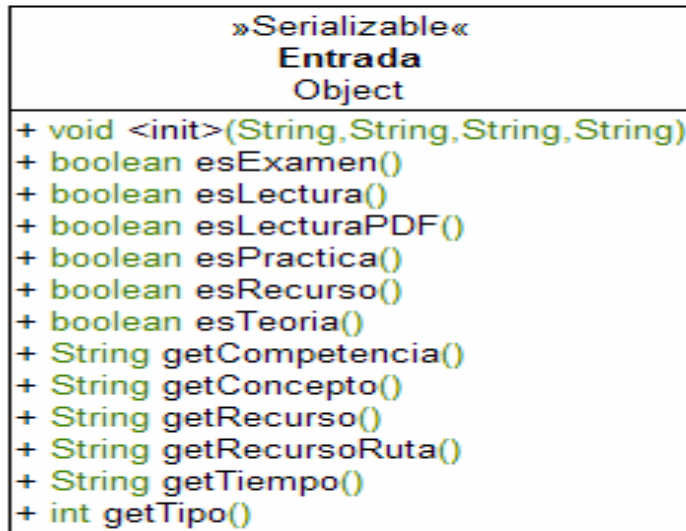


Figura 29: Módulo de Utilidades, Entrada

La clase Entrada es una de las clases más importantes en la lógica de la aplicación. Representa los recursos de los cuales se parte para realizar la composición de una unidad de aprendizaje, agrupando las características referentes a su competencia, su concepto, su recurso físico asociado y su tipo. En su definición aparecen todas las operaciones que van a permitir consultar el estado de la entrada, para que la aplicación pueda decidir que hacer con ellas.

El elemento entrada es uno de los que más va a ser utilizado por el usuario, ya que en función de los recursos que el usuario seleccione, se van a construir los objetos Entrada de los que se parte para poder componer un paquete de contenidos que cumpla el formato necesario para ser utilizado posteriormente en tareas de enseñanza por ordenador.

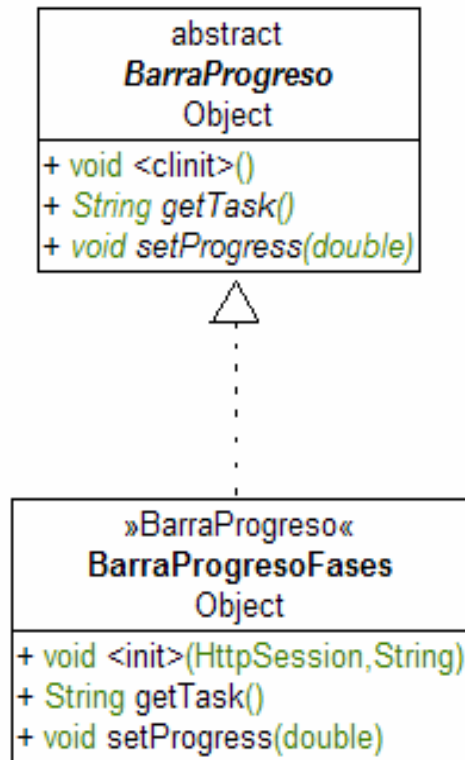


Figura 30: Módulo de Utilidades. Barra de Progreso

La interface **BarraProgreso** define un tipo de barra de estado para monitorizar un proceso. El elemento **BarraProgresoFases** implementa esta interface, y es el utilizado para monitorizar y mostrar al usuario el progreso del proceso de composición, en este caso a través de una barra de estado que se va rellenando de 0% a 100%, lo que permite que el usuario se mantenga informado en todo momento del estado en que se encuentra la aplicación en cada momento.

La implementación de esta forma permite que si el día de mañana se pretende monitorizar un proceso de forma distinta, se puede crear fácilmente una nueva, haciendo que se implemente la interface genérica e implementando la monitorización concreta en la nueva clase, lo que proporciona una gran capacidad de modificación futura a la aplicación con un impacto en el código leve.

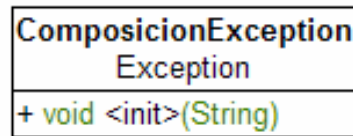


Figura 31: Módulo de Utilidades, Excepción

La clase CompositionException es una sencilla clase que define una excepción de error en el proceso de composición de unidades de aprendizaje, y que va a ser utilizada en el contexto de la aplicación para tener un mayor control de la gestión de errores de la misma en cada momento.

5.3.2.5. Módulo de Componentes Web

- **Nombre:** Módulo de Componentes Web.
- **Descripción general:** Este módulo contiene todos los servlets que se encargan de manejar el flujo de la aplicación.
- **Diagramas y funcionalidades específicas:**



Figura 32: Módulo de Componentes Web, Servlet Composición

El servlet Composición es el más importante de la aplicación, ya que es el encargado de recoger las entradas definitivas seleccionadas por el usuario y lanzar el

proceso de composición de la unidad de aprendizaje, controlando los posibles errores que se pudieran producir durante el mismo. Su acción va a derivar en la creación del material didáctico que corresponda al flujo de ejecución lanzada por el usuario en cada caso concreto.



Figura 33: Módulo de Componentes Web, Servlet VerRecurso

Este servlet es el que permite la visualización de los distintos recursos en el navegador en el que se está ejecutando la aplicación, ya sean html o pdf, así como la descarga de los archivos que él mismo seleccione. Es un servlet que va a permitir interactuar directamente con los recursos, por lo que es bastante importante para la aplicación.



Figura 34: Módulo de Componentes Web, Servlet ComposicionDescripciones

Este servlet se encarga de controlar el proceso de creación de los archivos que van a contener la introducción de la unidad de aprendizaje y las descripciones de las actividades que la forman, permitiendo que se muestre el formulario correspondiente una vez por cada una de las que compone la unidad, y que se termine el proceso cuando corresponda el momento.



Figura 35: Módulo de Componentes Web, Servlet Progreso

Este servlet es el encargado de lanzar el proceso de monitorización de la composición de la unidad de aprendizaje, controlando la presentación de la barra de progreso, así como el momento en el que debe ser cerrada una vez finalizado el proceso de composición, ya sea de manera exitosa o no.



Figura 36: Módulo de Componentes Web, Servlet AccionSiguiente

Este servlet se encarga de controlar el flujo de la aplicación una vez que se ha acabado la composición o ha ocurrido un error, permitiendo pasar a la siguiente fase o la vuelta al principio del proceso de composición, que es la fase de selección de recursos que cumplan las necesidades del usuario.



Figura 37: Módulo de Componentes Web, Filtro Composición

Este filtro se encarga de controlar el inicio del proceso de composición, lo que hace que, una vez que el usuario ha seleccionado los recursos que desea para que formen parte de la unidad de aprendizaje, el filtro decida si se puede comenzar o si los recursos que ha seleccionado no cumplen las condiciones necesarias para que se pueda realizar una composición de un paquete de contenidos correcta.

6. Manual de Configuración

Este apartado contiene la descripción de todos los archivos necesarios para la correcta configuración de la aplicación de Autoría de materiales didácticos. Los distintos archivos que son necesarios aparecen localizados y comentados de manera que se permitan posteriores modificaciones e los mismos si éstas fueran necesarias.

6.1. Mensajes de la aplicación

Los mensajes que la aplicación muestra en las diferentes pantallas que componen el flujo de ejecución vienen definidos en el archivo “*aplicacionComposicion.properties*”, que se encuentra en la carpeta “*includes*”, en la raíz del contenido Web de la aplicación (carpeta *WebContent*). Se detalla a continuación los mensajes y una vista de la pantalla en la que se pueden observar, pudiendo identificarse observando el texto y la pantalla para ver la situación de la propiedad en el entorno.

Los diferentes mensajes que contienen la aplicación son:

6.1.1. Mensajes mostrados en la pantalla de selección de recursos:

Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles					
Seleccionado	Competencia	Concepto	Tiempo	Recurso	Ver Descargar
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	Temas1y2.zip	 
<input type="checkbox"/>	comprender	Adapter	20 min	Tema3.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	10 min	practicaCVS.pdf	 
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	 
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	 
<input type="checkbox"/>	analizar	Command	10 min	practica10.pdf	 

Tipo de Unidad: Unidad de Aprendizaje basado en Problemas 
Componer

© 2009, CSS and XHTML

Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 38: Mensajes de la pantalla de selección de recursos

mensaje.inicioComposicion.recursos = Recursos disponibles
mensaje.inicioComposicion.verRecursoDisponible = Ver
mensaje.inicioComposicion.descargarRecursoDisponible = Descargar
mensaje.inicioComposicion.boton_ver = Visualizar el recurso
mensaje.inicioComposicion.boton_descargar = Descargar el recurso
mensaje.inicioComposicion.seleccionado = Seleccionado
mensaje.inicioComposicion.concepto = Concepto
mensaje.inicioComposicion.competencia = Competencia
mensaje.inicioComposicion.tiempo = Tiempo
mensaje.inicioComposicion.recurso = Recurso
mensaje.inicioComposicion.rutaUdC = Ruta de la Unidad creada:
mensaje.inicioComposicion.tipo = Tipo de Unidad:
mensaje.inicioComposicion.componer = Componer

En el desplegable de Tipo de Unidad:

mensaje.inicioComposicion.unidadPS = Unidad de Aprendizaje basado en Problemas
mensaje.inicioComposicion.unidadPE = Unidad de Aprendizaje basado en Procedimientos
mensaje.inicioComposicion.unidadSC = Unidad de Aprendizaje de Simple Colaboración

6.1.2. Mensajes mostrados en la cabecera de la pantalla y en la barra situada debajo de la cabecera de la pantalla:



Figura 39: Mensajes mostrados en la cabecera de la pantalla y en la barra situada debajo de la cabecera de la pantalla

mensaje.cabecera.textoInicial = Entorno de Autoría de Materiales Didácticos

mensaje.fasep.faseseleccion = Fase de selección de recursos
mensaje.fasep.fasecomposicion = Fase de composición
mensaje.fasep.faseexito = Fase de composición realizada con éxito
mensaje.fasep.fasepublicacion = Fase de publicación de material didáctico
mensaje.fasep.fasevisualizacion = Fase de visualización de material didáctico
mensaje.fasep.faseerrada = Error en la composición
mensaje.fasep.fasedescripciones = Fase de descripción de recursos

6.1.3. Mensajes mostrados en el pie de la pantalla:

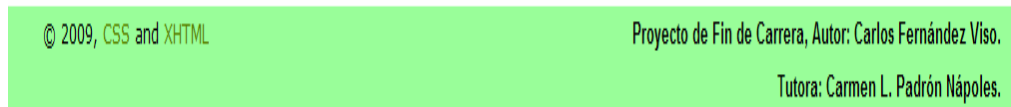


Figura 40: Mensajes mostrados en el pie de la pantalla

mensaje.pie.proyecto_fin_de_carrera = Proyecto de Fin de Carrera
mensaje.pie.autor = Autor: Carlos Fernández Viso
mensaje.pie.tutor = Tutora: Carmen L. Padrón Nápoles

6.1.4. Mensajes mostrados en la pantalla de introducción de descripciones de actividades:

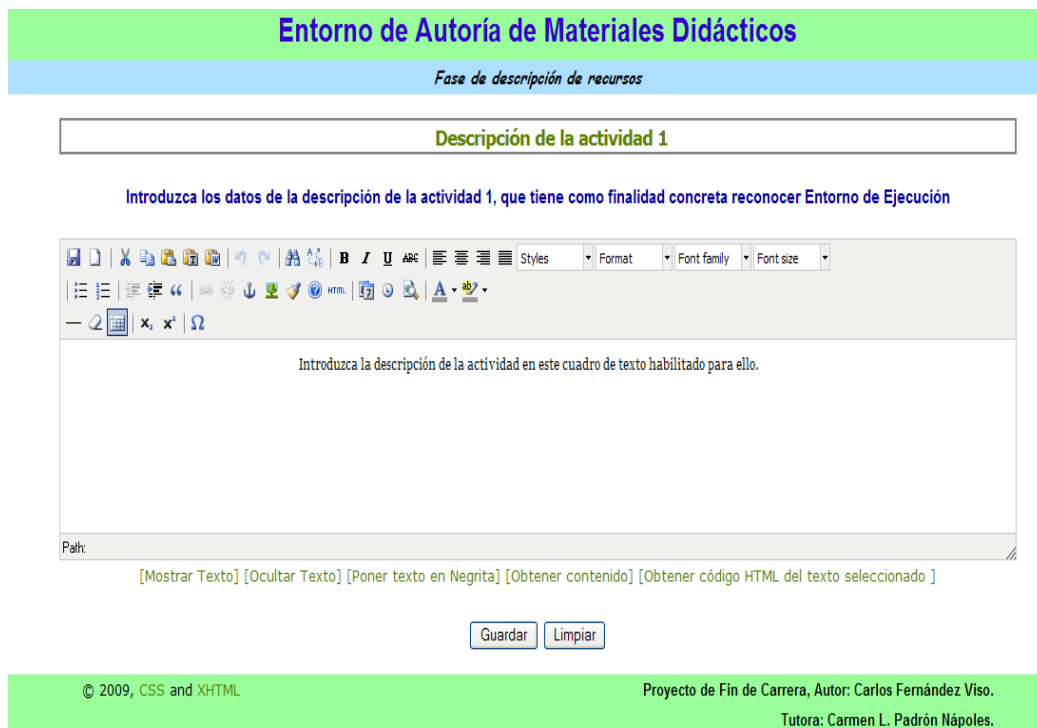


Figura 41: Mensajes mostrados en la pantalla de introducción de descripciones de actividades

mensaje.descripcion.introduccionActividad = Introduzca los datos de la descripción de la actividad
mensaje.descripcion.descripcionActividad = Descripción de la actividad
mensaje.descripcion.finalidad = que tiene como finalidad concreta
mensaje.descripcion.descripcion_inicial = Introduzca la descripción de la actividad en este cuadro de texto habilitado para ello.

mensaje.descripcion.Show = Mostrar Texto
mensaje.descripcion.Hide = Ocultar Texto
mensaje.descripcion.Bold = Poner texto en Negrita
mensaje.descripcion.Get_contents = Obtener contenido
mensaje.descripcion.Get_selected_HTML = Obtener código HTML del texto seleccionado
mensaje.descripcion.Get_selected_text = Obtener texto del elemento seleccionado
mensaje.descripcion.Get_selected_element = Obtener elemento seleccionado
mensaje.descripcion.guardar_descripcion = Guardar
mensaje.descripcion.limpiar_descripcion = Limpiar

6.1.5. Mensajes mostrados en la pantalla que indica el éxito en el proceso de composición:



Figura 42: Mensajes mostrados en la pantalla que indica el éxito en el proceso de composición

mensaje.composicion.tipo_unidad = Unidad de Tipo:
mensaje.composicion.creaciones = creada con éxito.
mensaje.composicion.volver_composicion = Volver a la página de composición
mensaje.composicion.publicacion_unidad_creada = Publicar
mensaje.composicion.descargarUnidad = Descargar la unidad creada

6.1.6. Mensajes que se muestran en la pantalla de error en el proceso de composición:



Figura 43: Mensajes que se muestran en la pantalla de error en el proceso de composición

mensaje.error.generales = Se ha producido un error en el proceso de composición.

mensaje.error.seleccion_recursos = Los recursos seleccionados no permiten la composición de una unidad de aprendizaje. Por favor, asegúrese de que ha seleccionado un mínimo de un recurso de teoría y que no ha seleccionado lecturas para un par de concepto y competencia para los que no se han seleccionado recursos de teoría asociados.

mensaje.error.redireccionado = por favor, pulse el siguiente enlace para volver a empezar el proceso.

mensaje.error.vuelta = Volver a la página de composición

6.2. Estilo visual de la aplicación

Los estilos que permiten modelar el aspecto visual de las distintas páginas WEB que componen la aplicación vienen definidos en el archivo `vista.css`, en la carpeta “`css`” del contenido Web de la aplicación. El hacer esto permite modificar los estilos de la aplicación sin tener que modificar el código fuente de las páginas, lo que favorece la posterior integración en un entorno con otros componentes WEB, y la modificación total o parcial de cualquier elemento concreto si es necesario realizarlo.

6.3. Repositorio de la aplicación

En la versión actual, el repositorio de recursos del que son leídos éstos para que se puedan incluir en el paquete de contenidos es parte integrante de la aplicación, es decir, todos los recursos que se utilizan están en la carpeta “`repositorio`”, con los demás componentes de la aplicación. La forma de implementar la composición permite que dicho repositorio esté definido en cualquier servidor WEB que sea accesible a través de una URL válida, pudiendo realizar el cambio sin necesidad de modificaciones del código fuente de la aplicación. Esto se logra porque se ha definido en el archivo “`aplicacionComposicion.properties`”, en la carpeta “`includes`” del contenido WEB de la aplicación, dos propiedades:

`url_repositorio = http://localhost:8080/EntAutoria/repositorio/`

Esta propiedad representa la dirección URL accesible vía WEB en la que están presentes los recursos que pueden ser utilizados en los procesos de composición derivados de la interacción de los usuarios con la aplicación.

`url_destino_compuesto=http://localhost:8080/EntAutoria/data/upload/`

Esta propiedad representa la ruta en la que se va a escribir el archivo zip de la unidad de aprendizaje creada, con todos los elementos que la forman. El hacer uso de esta propiedad permite que se pueda modificar para que se deje en un directorio remoto o en una carpeta de destino local, siempre y cuando dicho destino disponga de los permisos de escritura necesarios asociados al usuario con el que se ejecuta la aplicación WEB, ya que en caso de que esto no ocurra será imposible escribirlo en el lugar indicado. En estos momentos la aplicación escribe en la ruta de despliegue en el servidor de aplicaciones en el que se ejecuta, permitiendo la posterior descarga de las unidades siendo obtenidas de dicha ubicación determinada.

Es importante remarcar que las rutas deben llevar el símbolo “`/`” al final de la misma, puesto que en caso contrario la aplicación no tendrá un funcionamiento correcto.

6.4. Recursos obligatorios en las unidades

Los distintos tipos de unidades que se pueden componer a través de la aplicación tienen asociados una serie de recursos, tales como sus esquemas xsd, que van a ser fijos en todos los paquetes de contenidos que se obtengan como resultado de dicho procedimiento. Estos recursos se ubican en:

Unidades de aprendizaje basado en la resolución de problemas → en la carpeta “*data/plantillas/RecursosPE*” con el contenido WEB de la aplicación.

Unidades de aprendizaje basado en procedimientos → en la carpeta “*data/plantillas/RecursosPrE*” con el contenido WEB de la aplicación.

Unidades de aprendizaje colaborativo → en la carpeta “*data/plantillas/RecursosSC*” con el contenido WEB de la aplicación.

6.5. Despliegue de la aplicación WEB

6.5.1. JBoss

El despliegue de la aplicación WEB para que esté disponible para su utilización consiste en una serie de pasos, que son:

6.5.1.1. Windows

1. Colocar el archivo .war, que contiene la aplicación WEB, en el directorio de despliegue de JBoss, que es “*server\default\deploy*”. Asegurarse de que se encuentre en el directorio de despliegue del JBoss el ear de Coppercore (CopperCore.ear) para facilitar la publicación de las unidades creadas.

2. Poner en marcha el servidor de aplicaciones JBoss, ejecutando el script “*run.bat*” situado en el directorio “*bin*” de la carpeta jboss.

3. Una vez que ha salido por la consola todos los mensajes de log del servidor, hay que buscar que no muestre mensajes de fallo de despliegue en el contexto /EntAutoria, lo que indicaría que se ha producido un error que nos impedirá ejecutar la aplicación.

4. Comprobar que el servidor está arrancado , accediendo a “*http://{ip_servidor}:8080/*”, en donde ip_servidor es la dirección ip de acceso abierto de la máquina en la que se ejecuta el servidor. Si sale la consola de JBoss, el servidor se ha arrancado con éxito y se puede trabajar sin problemas con la aplicación desde este momento.

6.5.1.2. LINUX

1. Colocar el archivo .war, que contiene la aplicación WEB, en el directorio de despliegue de JBoss, que es “*server/default/deploy*”. Asegurarse de que se encuentre en el directorio de despliegue del JBoss el ear de Coppercore (CopperCore.ear) para facilitar la publicación de las unidades creadas.

2. Poner en marcha el servidor de aplicaciones JBoss, ejecutando el script “*run.sh*” situado en el directorio “*bin*” de la carpeta jboss.

3. Una vez que ha salido por la consola todos los mensajes de log del servidor, hay que buscar que no muestre mensajes de fallo de despliegue en el contexto /EntAutoria, lo que indicaría que se ha producido un error que nos impedirá ejecutar la aplicación.

4. Comprobar que el servidor está arrancado , accediendo a “*http://{ip_servidor}:8080/*”, en donde ip_servidor es la dirección ip de acceso abierto de la máquina en la que se ejecuta el servidor. Si sale la consola de JBoss, el servidor se ha arrancado con éxito y se puede trabajar sin problemas con la aplicación desde este momento.

7. Manual de Usuario

7.1. Visión general

El objetivo del presente documento es servir de guía para los usuarios de la aplicación de autoría de materiales didácticos, de manera que éstos puedan realizar todas las operaciones que el entorno permite y puedan maximizar el rendimiento que sacan de su interacción con la aplicación. En una primera fase este manual se presenta impreso, pero en un futuro se permitirá la consulta del mismo integrado en la aplicación a modo de html, existiendo en la actualidad la posibilidad de descargarse una copia del contenido de este documento en formato pdf(*“Portable Document Format”*) desde la propia aplicación.

Por cada una de las operaciones que puede efectuarse en el sistema, se mostrará una plantilla, compuesta por:

- **Nombre:** El nombre de la operación a realizar, es descriptivo de la funcionalidad que aporta.
- **Descripción general:** es un breve comentario de la acción de la funcionalidad.
- **Pantallas asociadas a la funcionalidad:** como su propio nombre indica, es el conjunto de pantallas con las cuáles el usuario puede visualizar lo que verá durante la ejecución de la aplicación. Para cada una se explica brevemente las posibilidades funcionales de los elementos que la componen.

7.2. Objetivos de la aplicación

El Entorno de Autoría de Materiales Didácticos es una aplicación destinada a facilitar el proceso de composición de paquetes de contenidos, también conocidos como unidades de aprendizaje, permitiendo de esta forma automatizar todo lo posible dicho proceso, permitiendo que los usuarios sin grandes conocimientos de informática y aspectos técnicos puedan hacer uso de las capacidades de la enseñanza a través de la web.

En la actualidad, la composición de materiales implicaba una serie de pasos que si bien para gente acostumbrada técnicamente a ello resultan triviales, para un usuario potencial de esta aplicación resultaban una tortura, ya que, por lo general, había que:

- Buscar todos los recursos necesarios para una unidad de aprendizaje, los cuales, en caso de no estar en un fichero .html o .pdf.
- Comprimir todos los ficheros en un archivo .zip y crear el archivo de manifiesto o descriptor de la unidad de aprendizaje, que sirve para describir la estructura del paquete, y que se trata de un fichero de xml, en el cuál hay que introducir los tag asociados de manera que puedan ser interpretados y sigan los estándares, que aunque actualmente no están implantados del todo, se está buscando dicho proceso, por lo que debe seguirse ese esquema.
- En caso de que alguno de los recursos no esté en un archivo .zip o pdf, hay que crear un archivo zip que sigue la estructura mencionada, con lo que es el doble de trabajo.
- Se da la circunstancia de que no todos los paquetes de contenidos son iguales, con lo que en caso de querer realizar la composición de paquetes de contenidos de diferentes tipos, habría que conocer los formatos de todos y cada uno de ellos, lo que multiplica la dificultad de forma excesiva.

Toda esta labor queda reducida a seleccionar los archivos que van a formar parte del paquete de contenidos, y realizar la composición, con lo que se obtiene un paquete de contenidos que lleva en su interior tanto el fichero de manifiesto como los recursos, organizados de forma que pueden ser interpretados por un visualizador de unidades de aprendizaje IMS LD adecuado para ello.

Carpeta		Carpeta	
..			
imgs		Carpeta	26/05/2008 13:55
articulos.html	1.198	517 Firefox Document	26/05/2005 13:27 5361EC3D
caracteristicas_POO.html	1.627	627 Firefox Document	21/05/2005 11:18 23C3AD35
componentesClase.html	1.575	753 Firefox Document	21/05/2005 11:18 6EF2BE1E
descripcion_clases.html	2.066	907 Firefox Document	28/05/2005 18:08 BBC49758
descripcion_conclusiones.html	2.247	966 Firefox Document	26/05/2005 14:50 C197B3F5
descripcion_ejemplos.html	1.842	855 Firefox Document	26/05/2005 13:24 E5C76C37
descripcion_objetos.html	1.791	843 Firefox Document	28/05/2005 18:08 6C06FADF
ejemplo_diseno.html	2.601	796 Firefox Document	21/05/2005 11:19 5846D0D6
ejemplo_diseno2.html	1.414	647 Firefox Document	23/05/2005 0:12 8BD75597
ejemplo_estructura_clase.html	1.358	614 Firefox Document	21/05/2005 11:19 5B130AC5
ejemplo_herencia_multiple.html	1.251	568 Firefox Document	21/05/2005 11:20 8E73A338
ejemplo1.html	1.933	747 Firefox Document	21/05/2005 11:19 F4A399F7
ejercicio_1.html	1.167	553 Firefox Document	21/05/2005 11:22 CBD283E9
ejercicio_1_respuesta.html	1.284	606 Firefox Document	21/05/2005 11:22 EC1ED484
ejercicio_2.html	1.183	560 Firefox Document	21/05/2005 11:21 C44CE405
ejercicio_2_respuesta.html	1.340	626 Firefox Document	21/05/2005 11:23 084DECD0
ejercicio_3_respuesta.html	1.342	644 Firefox Document	21/05/2005 11:23 D1120419
ejercicio_3.html	1.201	574 Firefox Document	21/05/2005 11:24 4BE563F8
encapsulamiento.html	2.162	971 Firefox Document	21/05/2005 11:24 2003BC24
glosario.html	2.037	940 Firefox Document	21/05/2005 11:24 924BCC8D
herencia.html	2.195	917 Firefox Document	23/05/2005 0:14 D599DF3F
IMS_LD_Level_A.xsd	26.958	3.457 XML Schema File	24/05/2005 13:21 02730F34
IMS_LD_Level_A_emaildata.xsd	1.485	546 XML Schema File	24/05/2005 13:21 A8FAA11F
IMS_LD_Level_A_reusables.xsd	1.784	581 XML Schema File	24/05/2005 13:21 EFD048D9
ims_xml.xsd	1.104	453 XML Schema File	24/05/2005 13:21 69DE7992

Figura 44: Ejemplo del interior de una unidad de aprendizaje

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This is a Reload version 2.0.2 Learning Design document-->
<!--Spawned from the Reload Learning Design Generator - http://www.reload.ac.uk-->
<manifest xmlns="http://www.imsglobal.org/xsd/imsdp_v1p1" xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xmlns:imsld="http://www.imsglobal.org/xsd/imsld_v1p0" identifier=
"MANIFEST-547653B5-8090-AC7D-1EB0-147574F36737" xsi:schemaLocation="http://www.imsglobal.org/xsd/imsdp_v1p1 imscp_v1p1.xsd
http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd http://www.imsglobal.org/xsd/imsld_v1p0 IMS_LD_Level_A.xsd">
  <organizations>
    <imsld:learning-design identifier="LD-3EE0C88D-49F2-681E-517C-35CB1A4D1D8D" level="A" uri="">
      <imsld:title>Curso de Programacion Orientada a Objetos</imsld:title>
      <imsld:learning-objectives>
        <imsld:title>Objetivos de aprendizaje</imsld:title>
        <imsld:item identifierref="RES-E6BF922E-FBA0-56EF-D3E9-76FD51A5840F" />
      </imsld:learning-objectives>
      <imsld:prerequisites>
        <imsld:title>Prerequisitos</imsld:title>
        <imsld:item identifierref="RES-436EAF98-5776-4712-246B-9F6A631C7E62" />
      </imsld:prerequisites>
      <imsld:components>
        <imsld:roles identifier="LD-7E197114-FCCF-5C5C-E70D-031E0E59B271">
          <imsld:learner identifier="LD-3AC53F69-6017-0452-6456-E18AE182C192">
            <imsld:title>Learner</imsld:title>
          </imsld:learner>
        </imsld:roles>
        <imsld:activities>
          <imsld:learning-activity identifier="LD-8B2DF6B1-D561-A695-41FF-74165F2F6AF2">
            <imsld:title>Objetivos de la realización del curso</imsld:title>
            <imsld:activity-description>
              <imsld:title>Objetivos de la realización del curso</imsld:title>
              <imsld:item identifierref="RES-E6BF922E-FBA0-56EF-D3E9-76FD51A5840F" />
            </imsld:activity-description>
            <imsld:complete-activity>
              <imsld:user-choice />
            </imsld:complete-activity>
          </imsld:learning-activity>
```

Figura 45: Ejemplo de archivo de manifiesto – parte 1 - Actividades

```
<imsld:learning-activity identifier="LD-41085C6E-4267-6D5E-34EE-2C9C5E8A390A">
  <imsld:title>Introducción 1</imsld:title>
  <imsld:activity-description>
    <imsld:title>Introducción 1</imsld:title>
    <imsld:item identifierref="RES-0BB4AFC1-A4D0-56C2-ECF9-8C0A7D9E841D" />
  </imsld:activity-description>
  <imsld:complete-activity>
    <imsld:user-choice />
  </imsld:complete-activity>
</imsld:learning-activity>
<imsld:learning-activity identifier="LD-21204309-A2C8-AD68-FFE2-5B1170987482">
  <imsld:title>Introduccion 2</imsld:title>
  <imsld:activity-description>
    <imsld:title>Introducción 2</imsld:title>
    <imsld:item identifierref="RES-077962E1-DF2A-6AB2-E9B0-8D8584177408" />
  </imsld:activity-description>
  <imsld:complete-activity>
    <imsld:user-choice />
  </imsld:complete-activity>
</imsld:learning-activity>
<imsld:learning-activity identifier="LD-77BC688A-ACCA-82C5-13FC-E82FD82AC786">
  <imsld:title>Descripcion Conceptos Clases</imsld:title>
  <imsld:activity-description>
    <imsld:title>Descripción conceptos Clases</imsld:title>
    <imsld:item identifierref="RES-8E172E58-B266-BCA3-BD5B-6C66731E4C43" />
  </imsld:activity-description>
  <imsld:complete-activity>
    <imsld:user-choice />
  </imsld:complete-activity>
</imsld:learning-activity>
<imsld:learning-activity identifier="LD-5387D51C-19FF-39E8-4E01-9FDCF833F277">
  <imsld:title>Componentes de una clase</imsld:title>
  <imsld:activity-description>
    <imsld:title>Componentes de una clase</imsld:title>
    <imsld:item identifierref="RES-FADC4A7D-4EF7-EE3F-911F-13C8A743E3CC" />
  </imsld:activity-description>
  <imsld:complete-activity>
```

Figura 46: Ejemplo de archivo de manifiesto – parte 2 - Actividades


```
</imsld:complete-activity>
</imsld:learning-activity>
<imsld:learning-activity identifier="LD-6DC8134F-8FF6-D621-75FB-B936F92ADEE8">
  <imsld:title>Estructura de una clase</imsld:title>
  <imsld:activity-description>
    <imsld:title>Estructura de una clase</imsld:title>
    <imsld:item identifierref="RES-BB4BC1DB-B20C-CEE0-1D2C-51DD0FEF1C3F" />
  </imsld:activity-description>
  <imsld:complete-activity>
    <imsld:user-choice />
  </imsld:complete-activity>
</imsld:learning-activity>
<imsld:learning-activity identifier="LD-72B51EDB-8446-2EE5-075F-9B6552D0D56E">
  <imsld:title>Ejemplo de una clase</imsld:title>
  <imsld:activity-description>
    <imsld:title>Ejemplo de una clase</imsld:title>
    <imsld:item identifierref="RES-1E82F125-0B31-B776-4157-19F17F224C9A" />
  </imsld:activity-description>
  <imsld:complete-activity>
    <imsld:user-choice />
  </imsld:complete-activity>
</imsld:learning-activity>
<imsld:learning-activity identifier="LD-21974690-F49F-06EF-F9E6-72747A891545">
  <imsld:title>Ejercicio 1</imsld:title>
  <imsld:activity-description>
    <imsld:title>Ejercicio 1</imsld:title>
    <imsld:item identifierref="RES-2C0BBFF0-5F62-7C65-86B0-5D8D8C9CE765" />
  </imsld:activity-description>
  <imsld:complete-activity>
    <imsld:user-choice />
  </imsld:complete-activity>
</imsld:learning-activity>
<imsld:learning-activity identifier="LD-ADEC7FAE-2F4D-A492-9E41-FDC820625917">
  <imsld:title>Descripcion conceptos objetos</imsld:title>
  <imsld:activity-description>
    <imsld:title>Descripción conceptos objetos</imsld:title>
    <imsld:item identifierref="RES-6A01CFCF-822B-D89C-C4B6-D8D7E70B7A9A" />
  </imsld:activity-description>
  <imsld:complete-activity>
    <imsld:user-choice />
  </imsld:complete-activity>
</imsld:learning-activity>
```

Figura 47: Ejemplo de archivo de manifiesto – parte 3 – Actividades

```

|         <imsld:activity-structure identifier="LD-F57C37D6-E9E2-5A3C-1520-BA1E31ACCE61" number-to-select="1" sort="as-is"
structure-type="selection">
  <imsld:title>Conclusiones</imsld:title>
  <imsld:environment-ref ref="LD-8ABE4DC1-78D5-85C8-2958-BA0E81658CFE" />
  <imsld:learning-activity-ref ref="LD-53AF18DE-5FEC-C3DD-62AE-8C119260B59C" />
</imsld:activity-structure>
<imsld:activity-structure identifier="LD-68C21901-2429-AA69-84F7-305C149D3F62" number-to-select="1" sort="as-is" structure-type=
"selection">
  <imsld:title>Test Evaluación</imsld:title>
  <imsld:learning-activity-ref ref="LD-234896C2-0F33-ADE7-1D62-615ABB9ADE11" />
</imsld:activity-structure>
</imsld:activities>
<imsld:environments>
  <imsld:environment identifier="LD-0B13E129-F34A-23D6-1065-F5E879A3517A">
    <imsld:title>Introducción a la POO</imsld:title>
    <imsld:learning-object identifier="LD-EBA08A0E-3960-1E7C-18AD-D6E8549383CF" type="knowledge-object">
      <imsld:title>Referencias</imsld:title>
      <imsld:item identifierref="RES-6545BB38-4774-4546-1A3D-03FCAA539562" />
    </imsld:learning-object>
  </imsld:environment>
  <imsld:environment identifier="LD-D44D99CC-6149-EF03-932C-97B451B3FB5C">
    <imsld:title>Conceptos principales en la POO</imsld:title>
    <imsld:learning-object identifier="LD-9E561DB1-DA32-56B9-C850-BFAE7137849F" type="knowledge-object">
      <imsld:title>Glosario</imsld:title>
      <imsld:item identifierref="RES-B8B47D0D-403B-EA1D-25FF-C28DFE5621F2" />
    </imsld:learning-object>
  </imsld:environment>
  <imsld:environment identifier="LD-E76EC7D1-371B-7C11-44D5-0DFA4713774E">
    <imsld:title>Conceptos complementarios en POO</imsld:title>
    <imsld:learning-object identifier="LD-89AC5EAC-CDB9-AB42-DE80-58DF73529F0C" type="knowledge-object">
      <imsld:title>Artículos sobre POO</imsld:title>
      <imsld:item identifierref="RES-3B87DEC3-7904-FB7E-4720-60F1D8DF7849" />
    </imsld:learning-object>
  </imsld:environment>
  <imsld:environment identifier="LD-07F2B086-CEA2-20F8-CE5E-77F688D2392E">
    <imsld:title>Ficheros concepto clase</imsld:title>
    <imsld:learning-object identifier="LD-2EB92D5C-13E0-75D6-22A0-3897EAD799E1" type="knowledge-object">

```

Figura 48: Ejemplo de archivo de manifiesto – parte 4 – Environments

```

<imsld:environment identifier="LD-0A63ECA9-5A13-A026-A21F-E4EA64012411">
  <imsld:title>Respuesta 3</imsld:title>
  <imsld:learning-object identifier="LD-52EA261E-DBD7-926C-C7DF-DB2940C6A267" type="knowledge-object">
    <imsld:title>Respuesta 3</imsld:title>
    <imsld:item identifierref="RES-78A3E9FA-025C-E2E8-DAA4-11AD655BAA6B" />
  </imsld:learning-object>
</imsld:environment>
</imsld:environments>
</imsld:components>
<imsld:method>
  <imsld:play identifier="LD-7B23F7B4-CBBF-5095-7460-C0529D2F574D">
    <imsld:title>Play</imsld:title>
    <imsld:act identifier="LD-E3AF610F-7E0D-79FD-7BE4-B6F55472AB39">
      <imsld:title>Acto 0</imsld:title>
      <imsld:role-part identifier="LD-FBD5F595-EA3F-29E6-632F-AAD7971435BE">
        <imsld:title>Role Part</imsld:title>
        <imsld:role-ref ref="LD-3AC53F69-6017-0452-6456-E18AE182C192" />
        <imsld:activity-structure-ref ref="LD-6E39B67B-BD86-8D43-ABFE-63E6D3348B87" />
      </imsld:role-part>
      <imsld:complete-act>
        <imsld:when-role-part-completed ref="LD-FBD5F595-EA3F-29E6-632F-AAD7971435BE" />
      </imsld:complete-act>
    </imsld:act>
    <imsld:act identifier="LD-A7F640D4-0D96-8EAF-2EB4-F53E2142A6D4">
      <imsld:title>Acto 1</imsld:title>
      <imsld:role-part identifier="LD-498CA52B-C7C8-CDBB-73B5-97B39BAA1F91">
        <imsld:title>Role Part</imsld:title>
        <imsld:role-ref ref="LD-3AC53F69-6017-0452-6456-E18AE182C192" />
        <imsld:activity-structure-ref ref="LD-148FE0A7-E7BD-2B83-043B-D7F8B1C4E0CD" />
      </imsld:role-part>
      <imsld:complete-act>
        <imsld:when-role-part-completed ref="LD-498CA52B-C7C8-CDBB-73B5-97B39BAA1F91" />
      </imsld:complete-act>
    </imsld:act>
    <imsld:act identifier="LD-2B775913-59DB-8596-36AD-3F07867758B3">
      <imsld:title>Acto 2</imsld:title>
      <imsld:role-part identifier="LD-8EDC742A-E9B9-6410-7CAD-57CE1FCC9EA3">

```

Figura 49: Ejemplo de archivo de manifiesto – parte 5 – Play

```
</imsld:learning-design>
</organizations>
<resources>
  <resource identifier="RES-3B87DEC3-7904-FB7E-4720-60F1D8DF7849" type="webcontent" href="articulos.html">
    <file href="articulos.html" />
  </resource>
  <resource identifier="RES-40059979-183D-85B9-F627-9A397CA9AF03" type="webcontent" href="caracteristicas_POO.html">
    <file href="caracteristicas_POO.html" />
  </resource>
  <resource identifier="RES-FADC4A7D-4EF7-EE3F-911F-13C8A743E3CC" type="webcontent" href="componentesClase.html">
    <file href="componentesClase.html" />
  </resource>
  <resource identifier="RES-8E172E58-B266-BCA3-BD5B-6C66731E4C43" type="webcontent" href="descripcion_clases.html">
    <file href="descripcion_clases.html" />
  </resource>
  <resource identifier="RES-F15A4235-3048-54A0-5BC7-09C177023507" type="webcontent" href="descripcion_conclusiones.html">
    <file href="descripcion_conclusiones.html" />
  </resource>
  <resource identifier="RES-BD262BCA-271A-3AB1-EBB3-41CC6CC40207" type="webcontent" href="descripcion_ejemplos.html">
    <file href="descripcion_ejemplos.html" />
  </resource>
  <resource identifier="RES-6A01CFCF-822B-D89C-C4B6-D8D7E70B7A9A" type="webcontent" href="descripcion_objetos.html">
    <file href="descripcion_objetos.html" />
  </resource>
  <resource identifier="RES-957BF4BC-72E6-B067-D4FF-D532E82AE2B9" type="webcontent" href="ejemplo_diseno.html">
    <file href="ejemplo_diseno.html" />
  </resource>
  <resource identifier="RES-2C0BBFF0-5F62-7C65-86B0-5D8D8C9CE765" type="webcontent" href="ejercicio_1.html">
    <file href="ejercicio_1.html" />
  </resource>
  <resource identifier="RES-2488F96A-1A60-7FDD-3333-ECE5D9E02463" type="webcontent" href="ejercicio_1_respuesta.html">
    <file href="ejercicio_1_respuesta.html" />
  </resource>
</resources>
```

Figura 50: Ejemplo de archivo de manifiesto – parte 6 – Recursos

En los pantallas anteriores, se ha podido observar la mayor parte de la estructura de un archivo de manifiesto, con lo que se ha podido observar la dificultad de su composición sin tener una herramienta para ello, por lo que se ve la necesidad y utilidad de la aplicación WEB desarrollada.

Para saber más sobre los ficheros imsmmanifest.xml su contenido y razón se aconseja que consulta el siguiente sitio Web:

<http://www.imsglobal.org/>

7.3. Guía de funcionalidades

7.3.1. Selección de Recursos

- **Nombre:** Selección de Recursos

















- **Descripción general:** El usuario puede, mediante esta funcionalidad, seleccionar los recursos que van a formar parte de la unidad de aprendizaje que se generará como resultado del uso de la aplicación.

- **Pantallas asociadas a la funcionalidad:**

Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles

Seleccionado	Competencia	Concepto	Tiempo	Recurso	Ver Descargar
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	Temas1y2.zip	 
<input type="checkbox"/>	comprender	Adapter	20 min	Tema3.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	40 min	practicaUMLJava.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	50 min	practicaAnt.tgz	 
<input type="checkbox"/>	comprender	Command	20 min	Tema4.zip	 
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	 
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	 
<input type="checkbox"/>	analizar	Factory	20 min	practica9.pdf	 

Tipo de Unidad: Unidad de Aprendizaje basado en Problemas ⓘ

Componer

© 2009, CSS and XHTML
Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

















Figura 51: Pantalla de Selección de Recursos (1)


En esta pantalla, el usuario puede marcar los diferentes recursos haciendo clic en las pestañas de la columna “*Seleccionado*” que acompañan a los distintos recursos, que aparecen acompañados de su “*Competencia*”, su “*Concepto*”, su “*Tiempo*”, que se corresponde con el tiempo estimado de duración del proceso de aprendizaje del mismo, el “*Recurso*”, que representa el nombre del archivo que forma el recurso descrito, y la columna de “*Ver/Descargar*”, que permite realizar esas acciones.

Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles

Seleccionado	Competencia	Concepto	Tiempo	Recurso	Ver Descargar
<input checked="" type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	Temasly2.zip	 
<input type="checkbox"/>	comprender	Adapter	20 min	Tema3.zip	 
<input checked="" type="checkbox"/>	construir	Entorno de Ejecución	40 min	practicaUMLJava.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	50 min	practicaAnt.tgz	 
<input checked="" type="checkbox"/>	comprender	Command	20 min	Tema4.zip	 
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	 
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	 
<input type="checkbox"/>	analizar	Factory	20 min	practica9.pdf	 

Tipo de Unidad: Unidad de Aprendizaje basado en Problemas 

Componer

© 2009, CSS and XHTML

Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 52: Pantalla de Selección de Recursos (2)


En esta pantalla puede verse como el usuario ha seleccionado 3 recursos, el “Temas1y2.zip”, cuya competencia es “reconocer” y su concepto es “Entorno de Ejecución”, “practicaUML_Java.zip”, cuya competencia es “construir” y su concepto es “Entorno de Ejecución”, y “Tema4.zip”, cuya competencia es “comprender” y el concepto asociado al recurso es “Command”.

En la siguiente pantalla, puede apreciarse como se realiza la selección del tipo de unidad que va a crearse como resultado del proceso de composición:

Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles

Seleccionado	Competencia	Concepto	Tiempo	Recurso	Ver Descargar
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	Temas1y2.zip	 
<input type="checkbox"/>	comprender	Adapter	20 min	Tema3.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	40 min	practicaUMLJava.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	50 min	practicaAnt.tgz	 
<input type="checkbox"/>	comprender	Command	20 min	Tema4.zip	 
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	 
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	 
<input type="checkbox"/>	analizar	Factory	20 min	practica9.pdf	 

Tipo de Unidad:

Unidad de Aprendizaje basado en Problemas
Unidad de Aprendizaje basado en Problemas
Unidad de Aprendizaje basado en Procedimientos
Unidad de Aprendizaje de Simple Colaboracion

© 2009, CSS and XHTML

Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 53: Pantalla de Selección de Recursos (3)

Como puede verse, la selección del tipo de unidad se realiza entre los tres tipos disponibles, “Unidad de Aprendizaje basado en Problemas”, “Unidad de Aprendizaje basado en Procedimientos”, y “Unidad de Aprendizaje de Simple Colaboración”.

En la parte derecha del desplegable que permite seleccionar el tipo de unidad puede observarse una imagen con una interrogación, dicha imagen, cuando se coloca el puntero del ratón encima permite mostrar información sobre el tipo de unidad que está seleccionado en ese momento en el apartado Tipo de Unidad, tal y como muestran las siguientes pantallas:

Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles

Seleccionado	Competencia	Concepto	Tiempo	Recurso	Ver Descargar
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	Temas1y2.zip	 
<input type="checkbox"/>	comprender	Adapter	20 min	Tema3.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	40 min	practicaUMLJava.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	50 min	practicaAnt.tgz	 
<input type="checkbox"/>	comprender	Command	20 min	Tema4.zip	 
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	 
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	 
<input type="checkbox"/>	analizar	Factory	20 min	practica9.pdf	 

Tipo de Unidad: Unidad de Aprendizaje basado en Problemas 

Componer

© 2009, CSS and XHTML

El aprendizaje de los estudiantes se basa en la resolución de un determinado problema relacionado con una temática específica. El problema se puede descomponer en pequeños problemas de manera que el estudiante vaya alcanzando progresivamente las habilidades, competencias y conocimientos relacionados con la temática.
 Tutora: Carmen L. Padron Napoles.

Figura 54: Pantalla de Selección de Recursos (4)

Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles

Seleccionado	Competencia	Concepto	Tiempo	Recurso	Ver Descargar
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	Temas1y2.zip	 
<input type="checkbox"/>	comprender	Adapter	20 min	Tema3.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	40 min	practicaUML.java.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	50 min	practicaAnt.tgz	 
<input type="checkbox"/>	comprender	Command	20 min	Tema4.zip	 
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	 
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	 
<input type="checkbox"/>	analizar	Factory	20 min	practica9.pdf	 

Tipo de Unidad: Unidad de Aprendizaje basado en Procedimientos 

Componer

Aprendizaje basado en procedimientos. Objetivo llegar a dominar conceptos, procesos y procedimientos, generalmente estables y relacionados con una temática determinada.

© 2009, CSS and XHTML

Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 55: Pantalla de Selección de Recursos (5)

Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles

Seleccionado	Competencia	Concepto	Tiempo	Recurso	Ver Descargar
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	Temas1y2.zip	 
<input type="checkbox"/>	comprender	Adapter	20 min	Tema3.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	40 min	practicaUMLJava.zip	 
<input type="checkbox"/>	construir	Entorno de Ejecución	50 min	practicaAnt.tgz	 
<input type="checkbox"/>	comprender	Command	20 min	Tema4.zip	 
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	 
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	 
<input type="checkbox"/>	analizar	Factory	20 min	practica9.pdf	 

Tipo de Unidad: Unidad de Aprendizaje de Simple Colaboracion 

Componer

Aprendizaje basado en procedimientos. Objetivo llegar a dominar conceptos, procesos y procedimientos, generalmente estables y relacionados con una temática determinada.

© 2009, CSS and XHTML
Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 56: Pantalla de Selección de Recursos (6)

Puede apreciarse en las pantallas anteriores los tres tipos de mensajes de descripciones que se muestran en función del tipo de unidad que esté seleccionado en el momento concreto.

Si el usuario lo desea, puede visualizar los recursos que sean archivos de tipo HTML y de tipo PDF, haciendo clic en el botón que viene representado con una lupa:

Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles

Seleccionado	Competencia	Concepto	Tiempo	Recurso	Ver Descargar
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	Temas1y2.zip	
<input type="checkbox"/>	comprender	Adapter	20 min	Tema3.zip	
<input type="checkbox"/>	construir	Entorno de Ejecución	40 min	practicaUMLJava.zip	
<input type="checkbox"/>	construir	Entorno de Ejecución	50 min	practicaAnt.tgz	
<input type="checkbox"/>	comprender	Command	20 min	Tema4.zip	
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	
<input type="checkbox"/>	analizar	Factory	20 min	practica9.pdf	

Tipo de Unidad: Unidad de Aprendizaje basado en Problemas

Componer

© 2009, CSS and XHTML
Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 57: Pantalla de Selección de Recursos (Visualización 1)

Si el usuario pulsa el botón, se abre una ventana en la que se puede visualizar el recurso, para que de esta manera pueda examinarlo antes de decidir si hace uso del mismo en el proceso de composición o lo descarta del mismo.

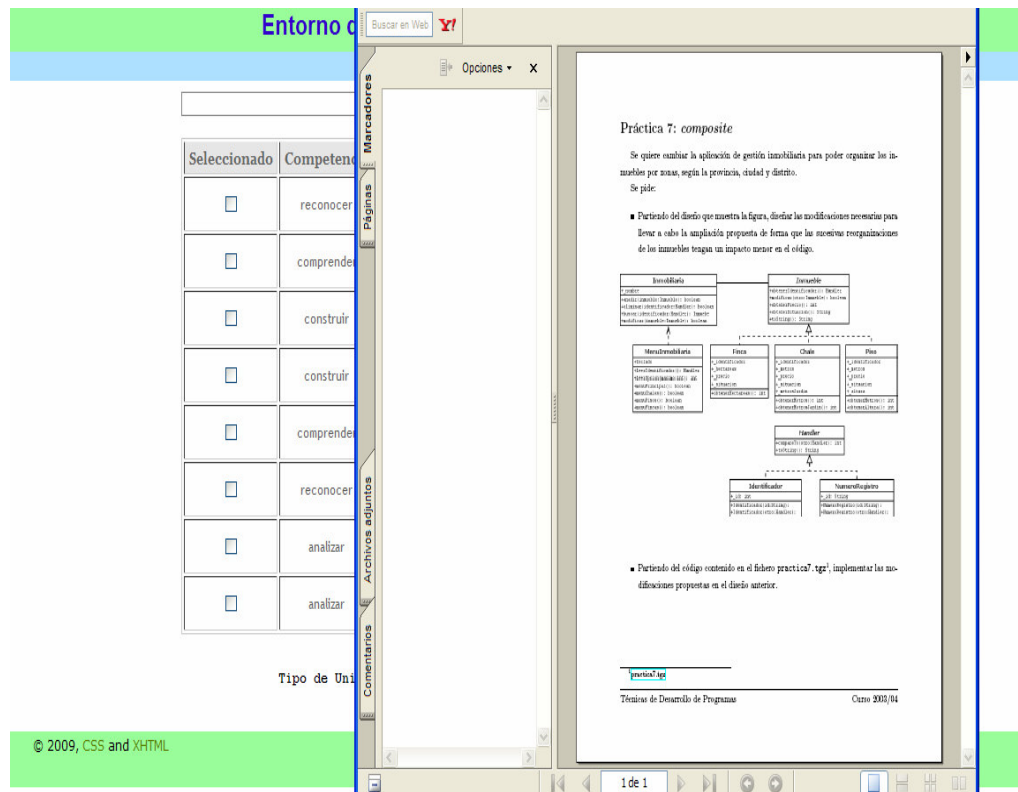


Figura 58: Pantalla de Selección de Recursos (Visualización 2)

Si lo desea, el usuario puede descargar el archivo a un directorio local que seleccione, para de esta manera poder analizarlo más profundamente y utilizarlo para lo que deseé, pudiendo de esta forma cerciorarse de que es el que desea que forme parte del paquete de contenidos que va a ser creado. Este proceso comienza abriendo una ventana, en la cuál el usuario debe decidir si quiere abrir el archivo o guardarlo en el equipo que está utilizando:

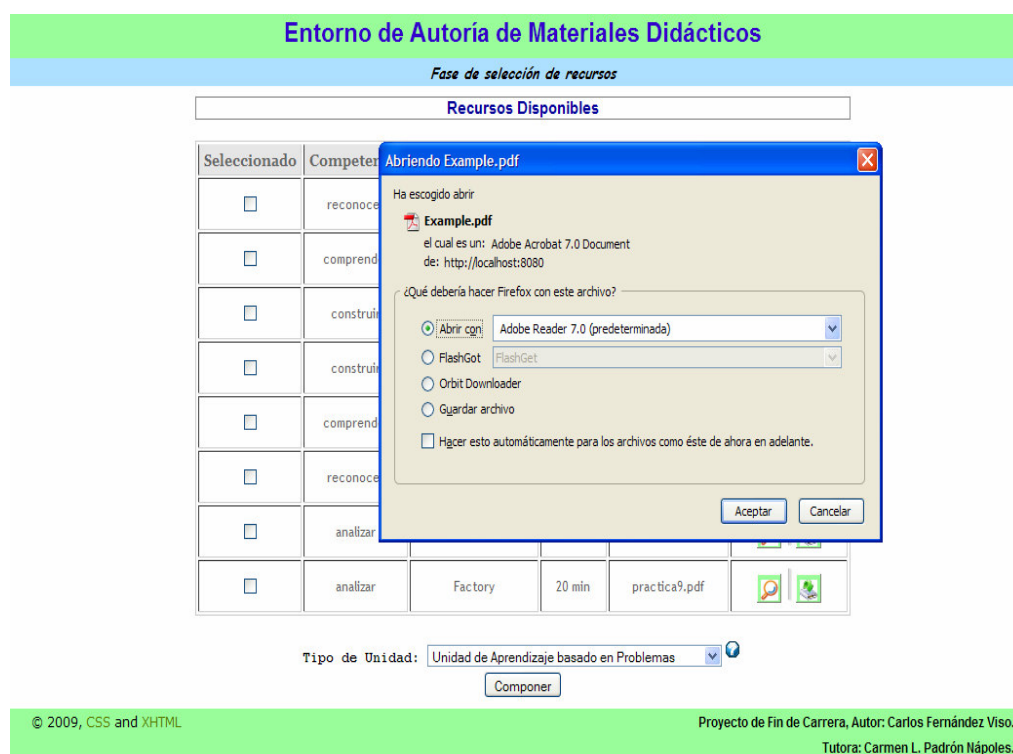


Figura 59: Pantalla de Selección de Recursos (Descarga 1)

El cuadro puede ser diferente en función del navegador que esté utilizando el usuario:

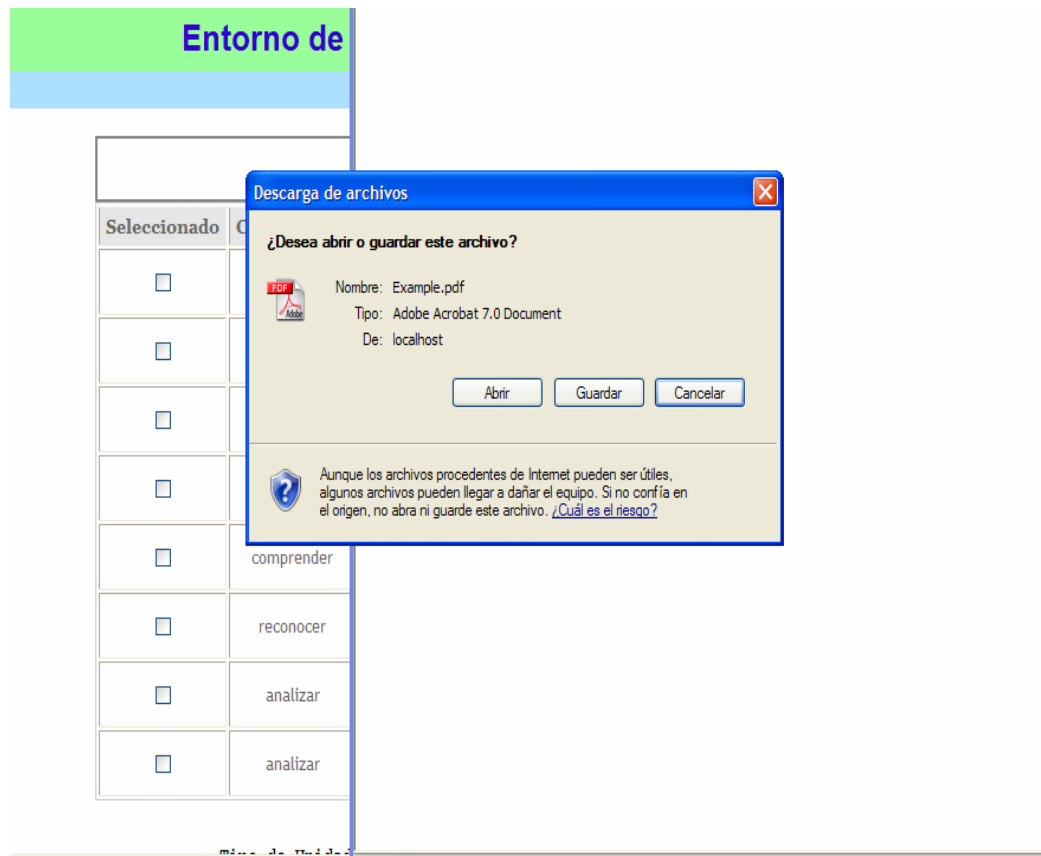


Figura 60: Pantalla de Selección de Recursos (Descarga 2)

En caso de tratarse de un archivo en formato .zip, la única posibilidad de que dispone el usuario es la descarga del archivo, debiendo de hacer uso de una programa adecuado para poder abrirlo y visualizar el contenido del recurso para examinarlo con detenimiento:

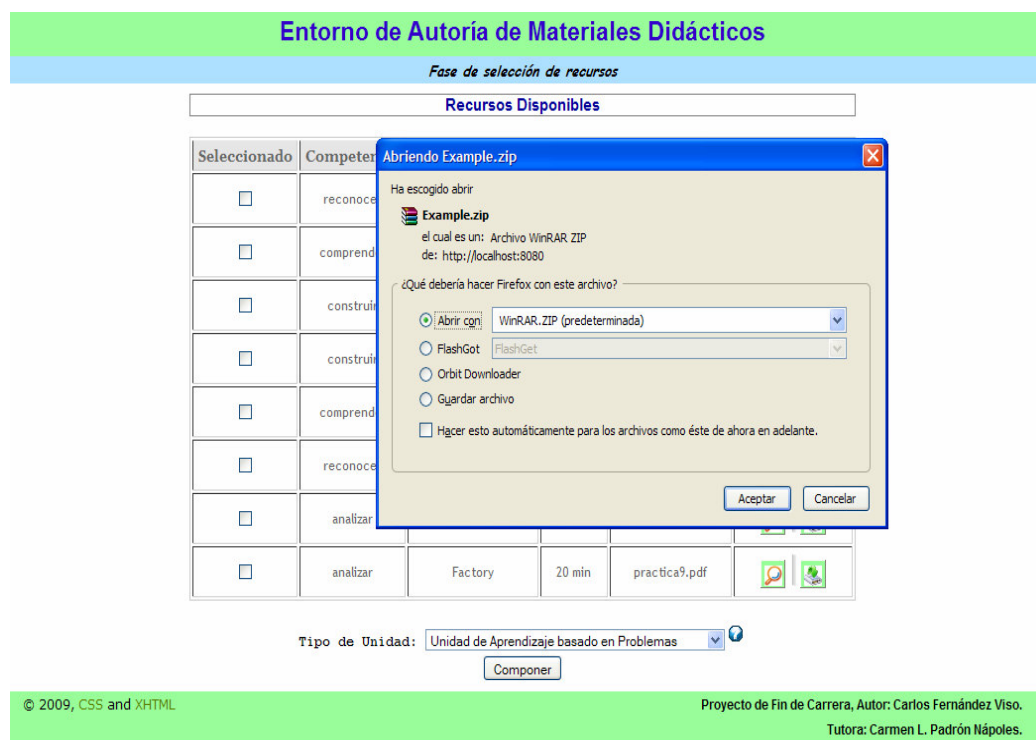


Figura 61: Pantalla de Selección de Recursos (Descarga 3)

















Si el usuario decide pasar a realizar la composición, se puede encontrar con varias pantallas en función de los datos de entrada que haya proporcionado, como son las siguientes:

- El usuario no ha seleccionado ningún recurso para realizar la composición:


Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles

Seleccionado	Competencia	Concepto	Tiempo	Recurso	Ver Descargar
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	Temasy2.zip	 
<input type="checkbox"/>	comprender	Adapter	20 min	Tema3.zip	 
<input type="checkbox"/>					 
<input type="checkbox"/>					 
<input type="checkbox"/>					 
<input type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	 
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	 
<input type="checkbox"/>	analizar	Factory	20 min	practica9.pdf	 

La página en http://localhost:8080 dice:

 Es necesario seleccionar un recurso como poco para poder realizar la composición

Aceptar

Tipo de Unidad: Unidad de Aprendizaje basado en Problemas

Componer

© 2009, CSS and XHTML

Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 62: Pantalla de Selección de Recursos (Error 1)

- El usuario ha seleccionado un grupo de recursos que no permite realizar un proceso de composición correcto, ya que no se pueden utilizar los que ha marcado para ello:



Figura 63: Pantalla de Selección de Recursos (Error 2)

Si el usuario desea volver al proceso de composición, ha de pulsar el botón “Volver a la página de composición”.

- El usuario ha seleccionado un grupo de recursos adecuado para realizar la composición, en este caso, la aplicación muestra una ventana con el progreso del proceso de composición, que se cerrará en el momento en el que éste llegue a su fin con éxito:

Entorno de Autoría de Materiales Didácticos

Fase de selección de recursos

Recursos Disponibles

Seleccionado	Competencia	Entorno de Ejecución	Tiempo	Recursos	Ver Descargar
<input checked="" type="checkbox"/>	reconocer			zip	
<input type="checkbox"/>	comprender			p	
<input type="checkbox"/>	construir			ava.zip	
<input type="checkbox"/>	construir	Entorno de Ejecución	50 min	practicaAnt.tgz	
<input type="checkbox"/>	comprender	Command	20 min	Tema4.zip	
<input checked="" type="checkbox"/>	reconocer	Entorno de Ejecución	10 min	practica7.pdf	
<input type="checkbox"/>	analizar	Command	10 min	practica8.pdf	
<input type="checkbox"/>	analizar	Factory	20 min	practica9.pdf	

Tipo de Unidad: Unidad de Aprendizaje basado en Problemas

Componer

© 2009, CSS and XHTML
Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 64: Pantalla de Composición (Barra de Progreso)

7.3.2. Introducción de descripciones de actividades

- **Nombre:** Introducción de descripciones de actividades.

- **Descripción general:** esta funcionalidad permite al usuario introducir las descripciones de las actividades que forman parte de la unidad, permitiendo de esta manera que pueda ser personalizable por su parte.

- **Pantallas asociadas a la funcionalidad:**

The screenshot shows a web-based interface for creating educational materials. At the top, a green header bar contains the title 'Entorno de Autoría de Materiales Didácticos'. Below it, a blue bar indicates the current phase: 'Fase de descripción de recursos'. A yellow box highlights the specific task: 'Descripción de la actividad 1'. The main area contains a blue instruction: 'Introduzca los datos de la descripción de la actividad 1, que tiene como finalidad concreta reconocer Entorno de Ejecución'. Below this is a rich text editor with a toolbar featuring icons for text formatting (bold, italic, underline, strikethrough, text color, background color), alignment, indentation, and other editing functions. The editor's text area contains the placeholder text: 'Introduzca la descripción de la actividad en este cuadro de texto habilitado para ello.' At the bottom of the editor, there is a 'Path:' label and a row of buttons: '[Mostrar Texto]', '[Ocultar Texto]', '[Poner texto en Negrita]', '[Obtener contenido]', and '[Obtener código HTML del texto seleccionado]'. A 'Guardar' (Save) button is located below the editor. The footer consists of a green bar with copyright information '© 2009, CSS and XHTML' on the left and project details 'Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso. Tutora: Carmen L. Padrón Nápoles.' on the right.

Figura 65: Introducción de descripciones de actividades (1)

En esta pantalla el usuario tiene el cuadro de texto para introducir el contenido que quiere que tenga el HTML que va a formar parte de la unidad de aprendizaje:

Entorno de Autoría de Materiales Didácticos

Fase de descripción de recursos

Descripción de la actividad 1

Introduzca los datos de la descripción de la actividad 1, que tiene como finalidad concreta reconocer Entorno de Ejecución

Introduzca la descripción de la actividad en este cuadro de texto habilitado para ello.

Path:

[Mostrar Texto] [Ocultar Texto] [Poner texto en Negrita] [Obtener contenido] [Obtener código HTML del texto seleccionado]

Guardar

© 2009, CSS and XHTML

Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 66: Introducción de descripciones de actividades (2)

En la parte superior del cuadro de texto, aparece indicado el número de la actividad actual, la competencia y el concepto de la misma, como puede verse en la siguiente pantalla señalizado:

Entorno de Autoría de Materiales Didácticos

Fase de descripción de recursos

Descripción de la actividad 1

Introduzca los datos de la descripción de la actividad 1, que tiene como finalidad concreta reconocer Entorno de Ejecución

Introduzca la descripción de la actividad en este cuadro de texto habilitado para ello.

Path:

[Mostrar Texto] [Ocultar Texto] [Poner texto en Negrita] [Obtener contenido] [Obtener código HTML del texto seleccionado]

Guardar

© 2009, CSS and XHTML

Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso.
Tutora: Carmen L. Padrón Nápoles.

Figura 67: Introducción de descripciones de actividades (3)

El usuario, además, tiene a su disposición una gran cantidad de controles para enriquecer el texto, permitiendo introducir texto en Negrita, en cursiva, subrayado...

The screenshot displays the 'Entorno de Autoría de Materiales Didácticos' (Authoring Environment for Didactic Materials) interface. At the top, a green header bar contains the title 'Entorno de Autoría de Materiales Didácticos'. Below it, a blue bar indicates the current phase: 'Fase de descripción de recursos'. The main content area is titled 'Descripción de la actividad 1'. It features a text input field with the placeholder text 'Introduzca los datos de la descripción de la actividad 1, que tiene como finalidad concreta reconocer Entorno de Ejecución'. Above the text field is a rich text editor toolbar with various icons for text formatting (bold, italic, underline, etc.) and alignment. Below the text field, there is a 'Path:' label and a row of buttons: '[Mostrar Texto]', '[Ocultar Texto]', '[Poner texto en Negrita]', '[Obtener contenido]', and '[Obtener código HTML del texto seleccionado]'. A 'Guardar' (Save) button is located at the bottom center. The footer consists of a green bar with copyright information '© 2009, CSS and XHTML' on the left and project details 'Proyecto de Fin de Carrera, Autor: Carlos Fernández Viso. Tutora: Carmen L. Padrón Nápoles.' on the right.

Figura 68: Introducción de descripciones de actividades (4)

Una vez que el usuario ha introducido los datos que quiere, le irá saliendo una pantalla que le permite introducir una descripción por cada una de las actividades que forman parte de la unidad de aprendizaje que se crea. Cuando ya ha terminado, pasa a la pantalla final del proceso de composición.

7.3.3. Finalización de la composición

- **Nombre:** Finalización de la composición

- **Descripción general:** El usuario, en el momento en el que ha completado el proceso de composición de una unidad de aprendizaje, selecciona lo que quiere hacer con el paquete de contenidos que ha creado, estado disponibles las opciones que pueden verse en la siguiente pantalla:

- **Pantallazas asociados a la funcionalidad:**



Figura 69: Pantalla de Finalización de la composición (1)

Si el usuario lo desea, puede descargarse la unidad de aprendizaje creada, haciendo clic en el botón “Descargar la unidad creada”, lo que le permitirá guardarla para su uso posterior o visualizar su contenido, en función de sus necesidades:



Figura 70: Pantalla de Finalización de la composición (2)

Otra opción del usuario es volver al principio de la composición, haciendo clic en el botón “Volver a la página de composición”, que aparece señalado en la siguiente pantalla:



Figura 71: Pantalla de Finalización de la composición (3)

Para finalizar el proceso, el usuario puede elegir publicar la unidad de aprendizaje, lo que le llevará a la dirección web del visualizador de paquete de contenidos SCORM, una funcionalidad que queda fuera del alcance de esta aplicación WEB, pero que será incorporada en una posterior integración de funcionalidades al procedimiento de composición de unidades de aprendizaje.

Si a lo largo de todo el proceso de composición se produce algún tipo de error, el usuario verá una pantalla como la que se muestra a continuación:



Figura 72: Finalización de la composición (4)

8. Gestión del Proyecto

La planificación del proyecto se ha realizado teniendo en cuenta una metodología en cascada y por tanto la realización de las tareas, en las que ha sido dividido el proyecto, se efectúan de manera sucesiva, es decir, cuando acaba una, comienza la siguiente. En el diagrama de tareas se muestra con detalle la división de tareas planteada para el proyecto, la duración de cada tarea, así como la fecha de comienzo y final de cada tarea.

El proyecto se inició en Septiembre de 2007 y concluyó en Julio de 2009, encontrándose mayores dificultades en las tareas de implementación de los módulos de visualización de contenidos y de composición de contenidos, aparte de períodos que por problemas externos al proyecto no se ha podido avanzar, lo cuál ha influido en la duración del mismo.

En la planificación del proyecto no se ha incluido ninguna gestión de riesgos, dado el carácter académico del mismo.

Las tareas llevadas a cabo son las siguientes:

- **Recogida de requisitos:** En esta tarea previa se han mantenido reuniones con la tutora del proyecto, que en este caso actuaba de cliente, para definir la funcionalidad que se deseaba que incluyera la aplicación. Esta fase ha sido de vital importancia para evitar futuros problemas derivados de una visión diferente de la aplicación por parte del cliente y de los desarrolladores.

- **Análisis:** En esta tarea se han realizado varias actividades, todas ellas comprendidas en la fase de análisis. Estas actividades son las necesarias para analizar el problema a partir de los requisitos de la fase previa. Como resultado de esta tarea se han obtenido diversos diagramas, como los productos de salida de las tareas: análisis de requisitos, definición de los casos de uso, casos de uso extendidos y análisis de otras herramientas, consiguiendo de esta forma modelar de forma visual los requisitos recogidos en la anterior fase de captura de requisitos.

- **Diseño:** Esta fase ha sido una de las más complicadas debido al hecho de que debía quedar marcado el camino que iba a tomar la implementación de la aplicación, pero todo ello capturando y plasmando las necesidades del cliente, modeladas en las anteriores fases. En esta fase se han realizado las siguientes tareas: realización del modelo lógico, realización del modelo de Clases, diagramas de secuencia, diagramas de objetos y diseño gráfico, que sirvieron de partida para realizar la implementación definitiva.

	i	Nombre de tarea	Duración	Comienzo	Fin
1	✓	Recogida de Requisitos	20 días	lun 01/10/07	vie 26/10/07
2	✓	☐ Análisis	34 días	vie 26/10/07	mié 12/12/07
3	✓	Análisis de Requisitos	14 días	vie 26/10/07	mié 14/11/07
4	✓	Definición de los casos de uso	8 días	jue 15/11/07	lun 26/11/07
5	✓	Casos de uso extendidos	5 días	mar 27/11/07	lun 03/12/07
6	✓	Análisis de otras herramientas	7 días	mar 04/12/07	mié 12/12/07
7	✓	Fin de Análisis	0 días	mié 12/12/07	mié 12/12/07
8	✓	☐ Diseño	28 días	vie 14/12/07	mar 22/01/08
9	✓	Modelo lógico	8 días	vie 14/12/07	mar 25/12/07
10	✓	Modelo de clases	10 días	mié 26/12/07	mar 08/01/08
11	✓	Diagramas de secuencia	3 días	mar 08/01/08	jue 10/01/08
12	✓	Diagramas de objetos	5 días	vie 11/01/08	jue 17/01/08
13	✓	Diseño gráfico	3 días	vie 18/01/08	mar 22/01/08
14	✓	Fin del Diseño	0 días	mar 22/01/08	mar 22/01/08
15	✓	☐ Implementación	76 días	mié 23/01/08	mié 07/05/08
16	✓	Módulo de Utilidades	18 días	mié 23/01/08	vie 15/02/08
17	✓	Modulo de Identificadores	10 días	sáb 16/02/08	vie 29/02/08
18	✓	Módulo de Unidades	20 días	sáb 01/03/08	vie 28/03/08
19	✓	Módulo de Fábricas de Unidades	12 días	sáb 29/03/08	mar 15/04/08
20	✓	Módulo de componentes WEB	16 días	mié 16/04/08	mié 07/05/08
21	✓	Fin Implementación	0 días	mié 07/05/08	mié 07/05/08
22	✓	Pruebas	6 días	sáb 17/05/08	mar 27/05/08
23	✓	Implantación	4 días	mié 28/05/08	lun 02/06/08
24	✓	Documentación del Proyecto	119 días	lun 05/01/09	jue 18/06/09

Figura 73: Planificación de tareas del Proyecto

- **Implementación:** En esta tarea se ha llevado a cabo toda la codificación de la aplicación. Se siguió una metodología de codificación por módulos, de manera que primero se realizaba la implementación de un módulo, se realizaban pruebas, y luego se pasaba a desarrollar el siguiente, pasando por el mismo proceso. El producto de salida de esta fase es el código definitivo de la aplicación WEB.

- **Pruebas:** En esta fase se realizaron pruebas individuales de todas y cada una de las funcionalidades de los módulos de la aplicación de forma exhaustiva, permitiendo de esta forma estar seguro de que los módulos tenían la funcionalidad exigida, y posteriormente se realizó una serie de pruebas de integración que permitieron comprobar que la acción conjunta de los módulos daba como resultado el esperado y deseado para satisfacer las necesidades del cliente en todo momento.

- **Implantación:** En esta tarea se han llevado a cabo las acciones necesarias para poner en funcionamiento en un entorno de producción la aplicación. Para ello se ha configurado el servidor de aplicaciones correctamente y se ha desplegado la aplicación WEB en el mismo, permitiendo que quede preparada para su utilización.

- **Documentación del proyecto:** Para finalizar el trabajo del proyecto se ha realizado este documento, recopilando para ello toda la información y documentación generada a lo largo del mismo, y siguiendo el estándar ESA [24], detallando todos y cada uno de los aspectos considerados necesarios para comprender su alcance y funcionamiento.

En la siguiente figura se muestra el diagrama Gantt relativo al proyecto, en él se pueden ver las diferentes tareas realizadas y el tiempo empleado en cada una de ellas, mostrando a su vez las relaciones existentes entre las diferentes tareas.

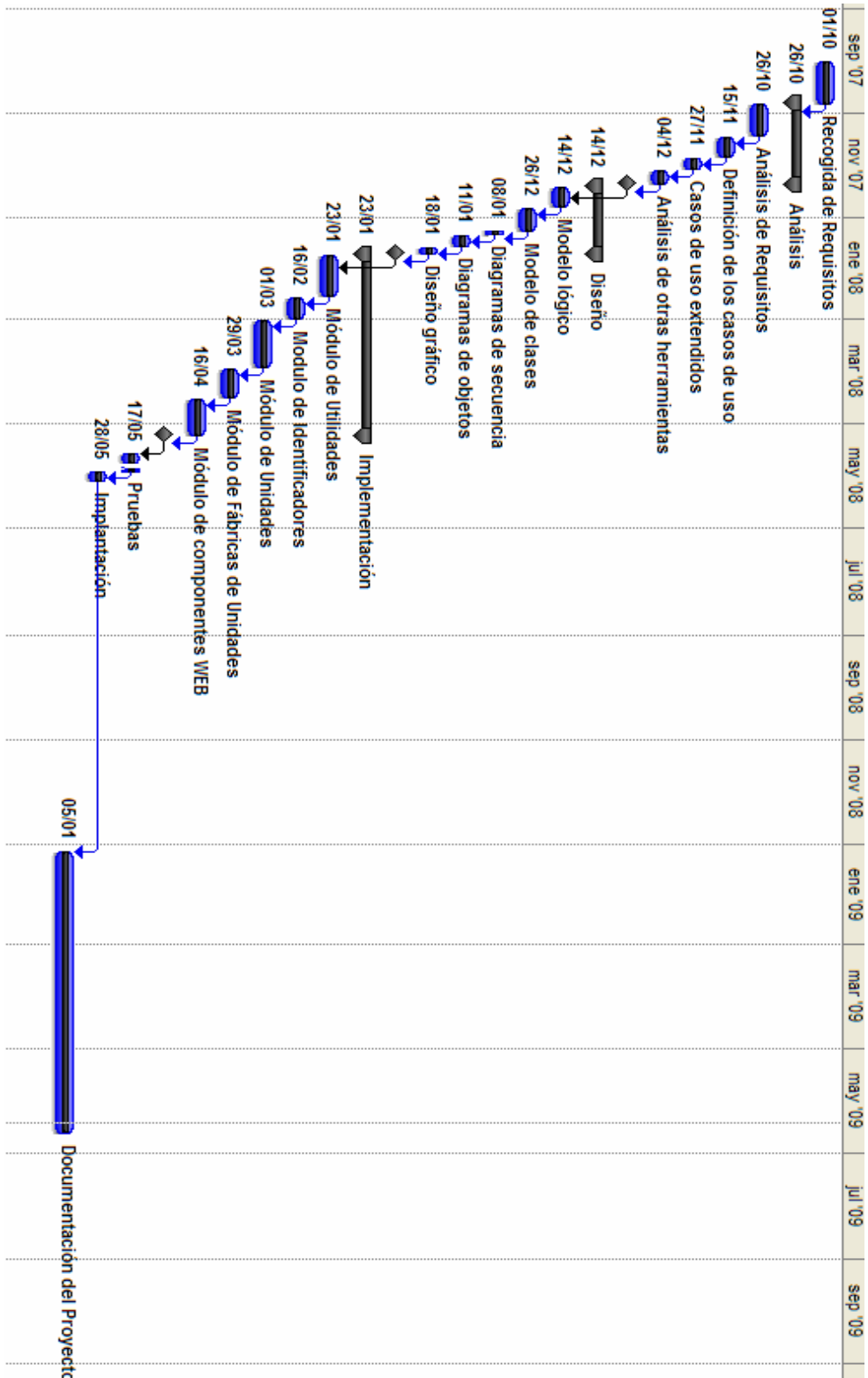


Figura 74: Diagrama Gantt del Proyecto

8.1. Análisis de Costes

Para analizar los costes que ha supuesto el desarrollo de la aplicación hemos dividido estos en grupos, quedando éstos organizados de la siguiente manera: personal implicado, material e infraestructuras utilizadas y otros. De esta manera, queda más claro en que parte se ha invertido el dinero necesario para el desarrollo de la aplicación.

8.1.1. Personal

Para el desarrollo de la aplicación han sido necesarios varios perfiles de personal, entre ellos un analista funcional, un analista orgánico y un analista/programador. Se ha prescindido del perfil de Jefe de proyecto ya que realmente el equipo de trabajo solo constaba de una persona debido al carácter académico del proyecto y sus tareas se han agrupado dentro del mismo perfil al analista funcional y al analista orgánico. Por lo tanto para analizar los costes de personal solo se han tenido en cuenta dos perfiles: el analista funcional (que también realizará las tareas de analista orgánico) y el analista/programador.

A continuación detallamos los días que cada perfil ha trabajado en cada actividad para el desarrollo final de la herramienta:

Tarea/Número de días	Analista Funcional	Analista Programador
Recogida de Requisitos	20	0
Análisis	34	0
Análisis de Requisitos	14	0
Definición de los casos de uso	8	0
Casos de uso extendidos	5	0
Análisis de otras herramientas	7	0
Diseño	24	16
Modelo lógico	6	2
Modelo de clases	8	4
Diagramas de secuencia	3	3
Diagramas de objetos	4	4
Diseño gráfico	3	3
Implementación	0	106
Módulo de Utilidades	0	24
Modulo de Identificadores	0	16
Módulo de Unidades	0	22
Módulo de Fábricas de Unidades	0	16
Módulo de componentes WEB	0	28
Pruebas	10	10
Implantación	0	2
Documentación del Proyecto	51	68
TOTAL	139	202

Figura 75: Tiempos del personal implicado en el Proyecto

El resumen de días y costes totales se muestra en la siguiente tabla. El precio de la hora se ha calculado en función de la estimación de diferentes proyectos públicos [25]:

	Días trabajados	€uros/Día	Coste Total
Analista Funcional	139	120 €	16.680 €
Analista Programador	202	80 €	16.160 €
TOTAL:			32.840 €

Figura 76: Resumen de días y costes totales del personal del Proyecto

8.1.2. Material e Infraestructuras

Para el desarrollo de la aplicación también ha sido necesario el uso de distintos tipos de material, desde el hardware utilizado para el desarrollo de la aplicación WEB y sus posteriores pruebas hasta elementos como el papel que ha hecho falta para todas las notas tomadas, los dispositivos de almacenamiento externos como CD o USB o la conexión a Internet. Por ello he realizado la siguiente tabla para indicar los costes de los mismos:

	Precio	Cantidad	SubTotal
Ordenador portátil HP Pavilion	2000 €	1	2000 €
Licencia de Microsoft Windows XP	132€	1	132 €
CD-R 80 min.	0,50 €	4	2 €
Papel A-4 (500 hojas)	3 €	1	3 €
Tinta impresora	30 €	2	60 €
PEN-Drive 2 GB	16 €	1	16 €
Conexión a Internet	35 €	12	420 €
IVA 16%			421,12 €
TOTAL			3053,12 €

Figura 77: Costes de materiales e infraestructuras

8.1.3. Otros

Por último, tendríamos los costes que no se podrían englobar en ninguno de los apartados anteriores y que se agrupan en este apartado. Los gastos relevantes se refieren a la gasolina, ya que ha sido necesaria para poder desplazarse a las reuniones con el cliente. Siguiendo la misma metodología que con los anteriores, se puede ver la siguiente tabla para visualizar estos costes:

	Precio	Cantidad	SubTotal
Gasolina Diesel (80 litros)	64 €	1	64 €
TOTAL			64 €

Figura 78: Otros gastos

8.1.4. Resumen de Costes

Para finalizar con el análisis de costes, se adjunta una tabla en la que se puede ver un resumen de los mismos, de tal manera que se contabilizan los gastos totales que ha supuesto el desarrollo de la aplicación que nos ocupa:

Tipo de gasto	Coste Total
Personal	32.840 €
Material e Infraestructura	3.053,12 €
Otros Gastos	64 €
TOTAL	35.957,12 €

Figura 79: Resumen de costes del Proyecto

Para finalizar, y realizando una apreciación que estima que pretendemos obtener en torno al 20% de beneficio, se le suma al total obteniendo la cifra que se le va a cobrar al cliente en este caso:

$$35.957,12 \text{ €} + 7.194,42 \text{ €} = 43.151,54 \text{ €}$$

9. Conclusiones

9.1. Evaluación

La aplicación **UoLComposer** cuyo proceso de desarrollo se explica en este documento ha sido utilizada en las pruebas de evaluación de la solución **MD2: Método de desarrollo de materiales didácticos** [26]. Específicamente se utilizó en las pruebas para la valoración de la utilidad del método para guiar el desarrollo. Esta evaluación de carácter empírico, tuvo como objetivo comprobar la capacidad del método propuesto en MD2 para guiar el proceso de desarrollo y si éste facilitaba soluciones adecuadas en las diferentes etapas del desarrollo. En el escenario de evaluación se utilizó la definición del método de desarrollo, implementaciones de prototipos de la herramienta basada en la solución propuesta: **UoLComposer** (descrita en este documento) y QTIComposer [27] y las herramientas de autoría RELOAD, Sled y QTeditor carentes de método de desarrollo. Los participantes fueron estudiantes que cursaban la asignatura “Enseñanza asistida por ordenador” del 5to año de la Titulación Ingeniería Informática de la Universidad Carlos III de Madrid en los cursos 2007-2008 y 2008-2009. Como mecanismos de recopilación de datos se emplearon encuestas a participantes, el contenido de dichas encuestas se presenta en el anexo 1. Los casos de estudio fueron dos situaciones de desarrollo de diferentes tipos de materiales, en las que los participantes tienen diferentes niveles de conocimientos de sobre las especificaciones: novel y medio. Éstos individuos se clasificaron en dos tipos: profesionales tipo medio, es decir, aquellas personas que poseían ciertos conocimientos sobre los estándares y especificaciones e-Learning, y tipo novel: aquellos que tenían contacto por primera vez con los estándares y especificaciones. Se realizaron 4 pruebas cuya descripción general se presenta en la Tabla de la Figura 80. La aplicación **UoLComposer** se utilizó en las pruebas IV.

Prueba	Curso/Participantes	Situación de desarrollo	Tipo de estudiante	Soporte del método	Enunciado
I	2006-2007 16 participantes	I (IMS LD)	NOVEL	NO	P6LD2006-07 (Anexo 1)
II		II(IMS LD+QTI+CP)	MEDIO		POblig2006-07 (Anexo 1)
III	2007-2008 6 participantes 2008-2009 6 participantes	I (IMS LD)	NOVEL	SI	P6LD2007-08 (Anexo 1)
IV		II(IMS LD+QTI+CP)	MEDIO		POblig200708 (Anexo 1)

Figura 80: Detalles de las pruebas realizadas para la evaluación del método propuesto por MD2.

Los datos obtenidos en las pruebas, mostrados en las Figuras 81 a 87, demuestran que el método es capaz de guiar el desarrollo, al ofrecer guías y medios para resolver las diferentes etapas del desarrollo. El soporte para la etapa de selección es idóneo puesto que se ofrecen guías y medios para poder elegir los recursos de contenido y estrategia pedagógica y en caso de ser necesario, crearlos, adaptarlos o modificarlos para que se ajusten a los requisitos de las situaciones de desarrollo. En el caso de la etapa de composición, el soporte del método es apropiado puesto que se ofrecen los medios (plantillas de la estructura) y mecanismos para la integración de los recursos en la estructura de presentación y entrega, así como para la visualización y comprobación de la conformidad con la especificación y correcto funcionamiento del material como paso previo indispensable para la evaluación. Mientras que los datos obtenidos demuestran la necesidad disponer de criterios y procedimientos claramente especificados para llevar a cabo la evaluación. Dichos elementos están definidos en el método y la arquitectura propuesta por la solución pero no estaban implementados en los prototipos utilizados.

Resultados de las Pruebas de Evaluación

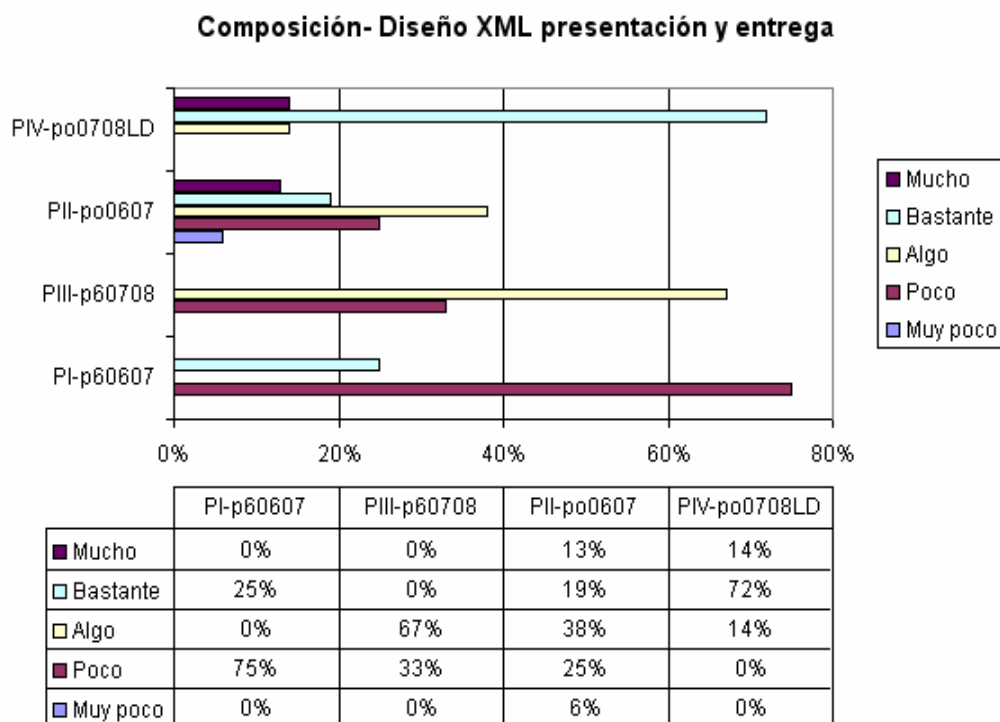


Figura 81: Resultados asociados al proceso de Composición-Diseño XML

En la tabla, en donde la aplicación queda identificada como PIV-po0708LD, se puede observar que la ayuda al proceso de composición y diseño de XML es calificada en su mayoría por los usuarios como mucha y bastante, lo que en comparación con las demás herramientas es una mejora considerable, ya que con ellas este procedimiento era costoso y se proporcionaba poca ayuda al mismo.

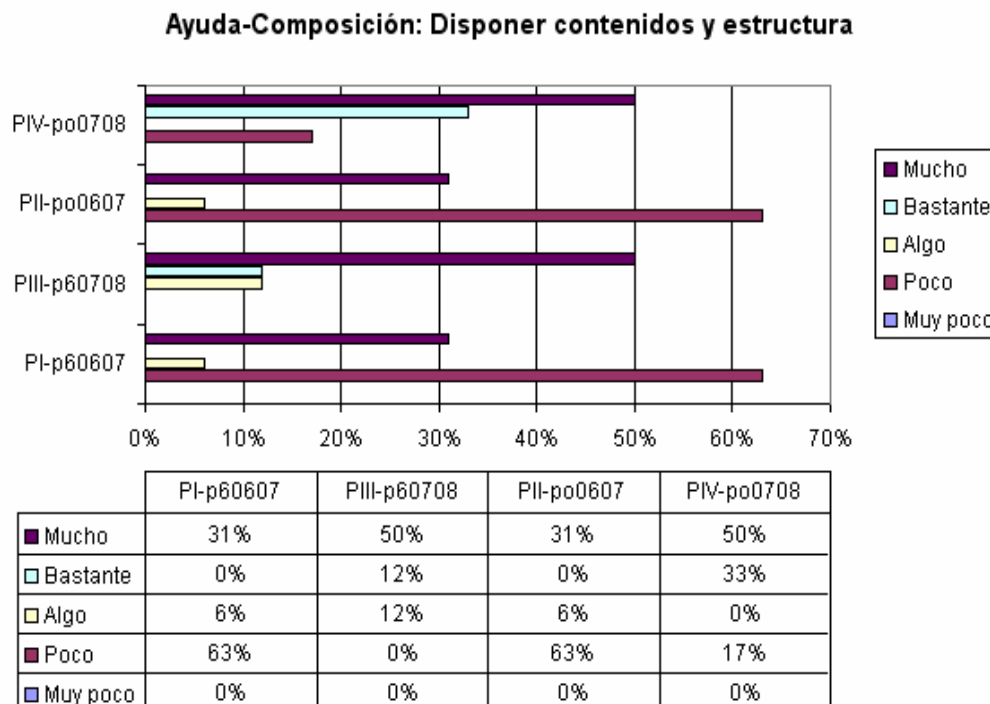


Figura 82: Resultados asociados a la disposición de contenidos, estrategia y estructura

En la tabla, en donde la aplicación queda identificada como PIV-po0708, se puede observar que la ayuda al proceso de composición, en lo referente a la disposición de contenidos y estructura, es calificada en su mayoría por los usuarios como mucha y bastante, ya que el proceso queda totalmente automático para ellos, que seleccionan los recursos que consideran que deben aparecer en el material y se olvidan del resto del proceso, lo que en contrapartida a las demás herramientas es una mejora importante, siendo un procedimiento excesivamente complejo en las otras herramientas utilizadas en las pruebas de evaluación.

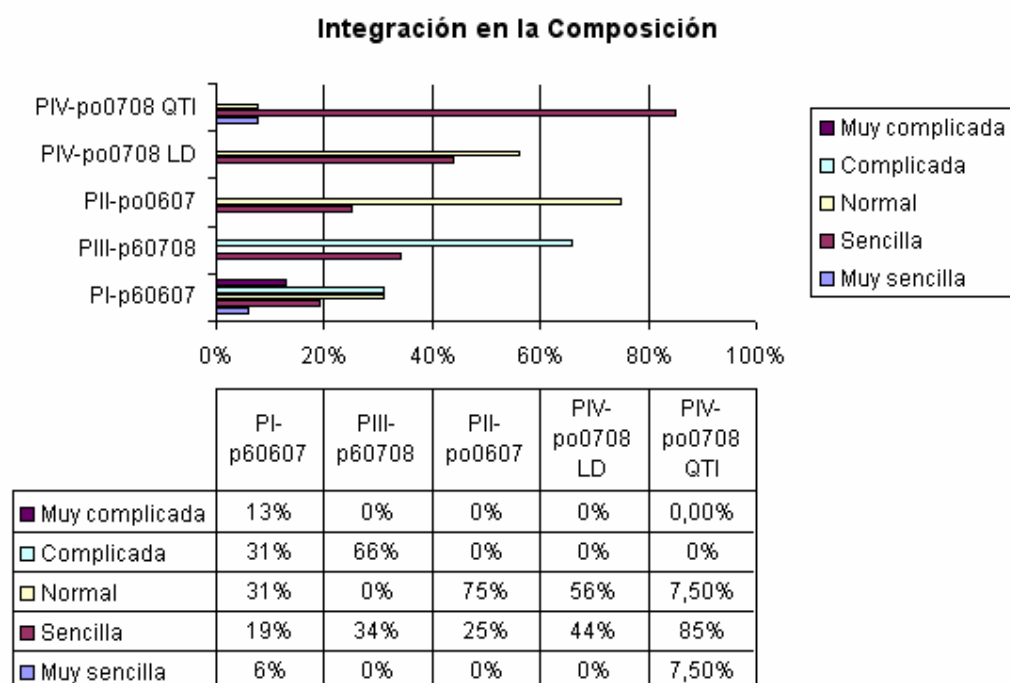


Figura 83: Resultados asociados al proceso de integración

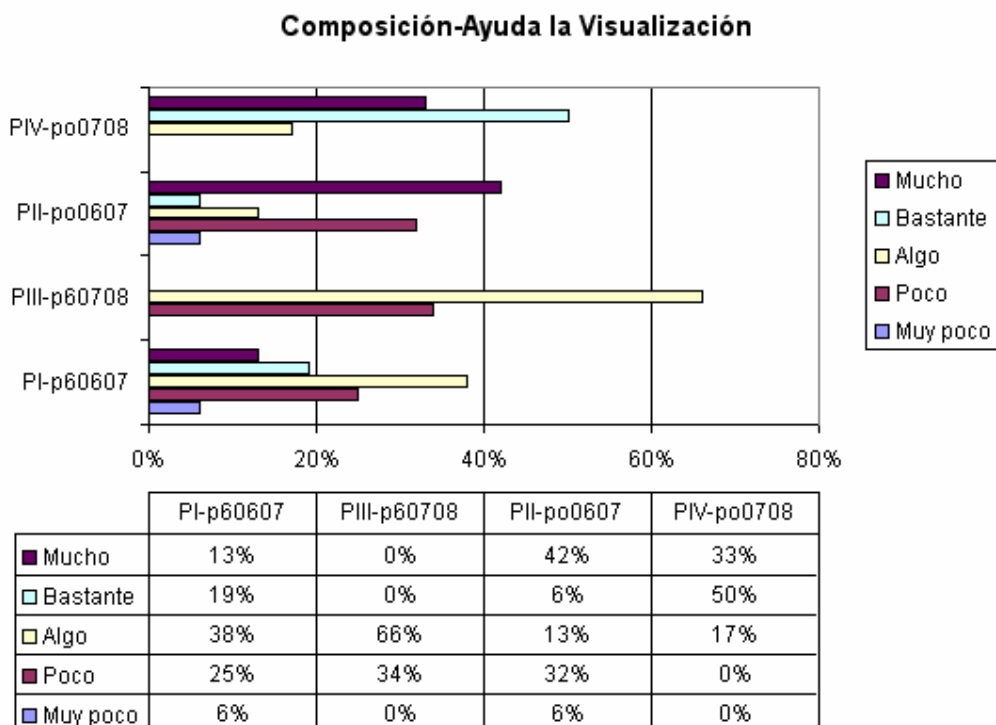


Figura 84: Resultados asociados a la ayuda a la visualización

En las tablas anteriores, se ha podido apreciar como en lo referente a la integración en la composición y en la ayuda a la visualización, la herramienta, PIV-po0708LD, ofrece una mejora considerable con respecto a las demás, salvo en el caso de la herramienta de composición de elementos QTI, con el que la mejora no es tan relevante, pero esto es comprensible puesto que se trata de un tipo de unidad específica más sencilla y que ofrece muchas menos variantes, por lo que se entiende que una aplicación bien diseñada para un problema más sencillo resulte más sencilla de utilizar, no por ello perdiendo el punto de vista con respecto a las otras herramientas previas al ámbito del proyecto, las cuáles no proporcionaban ninguna ventaja al usuario de la misma.

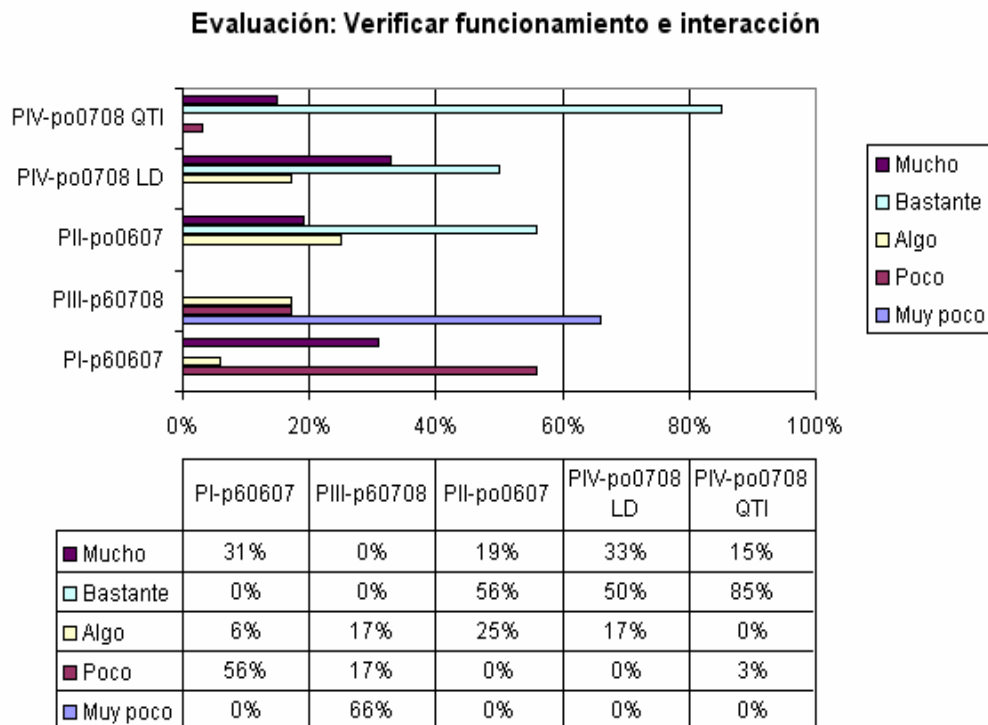


Figura 85: Resultados del proceso de evaluación

En la tabla se puede observar la importante cantidad de usuarios que califican la verificación del funcionamiento y la interacción como muy buena y bastante buena, lo que destaca el hecho de que la herramienta es muy buena en esos aspectos y por lo tanto cumple los objetivos establecidos de mejorar la integración con motores de publicación y visualización de paquetes de contenidos IMS LD. Estos datos quedan ampliamente remarcados en las siguientes tablas, donde se puede apreciar que el tiempo empleado en el desarrollo de nuevos paquetes de contenidos y la ayuda que se muestra al usuario mejora espectacularmente respecto a las demás, se puede comprobar observando los datos de PIV-p0708LD en las gráficas adjuntas.

Desarrollo con las herramientas analizadas

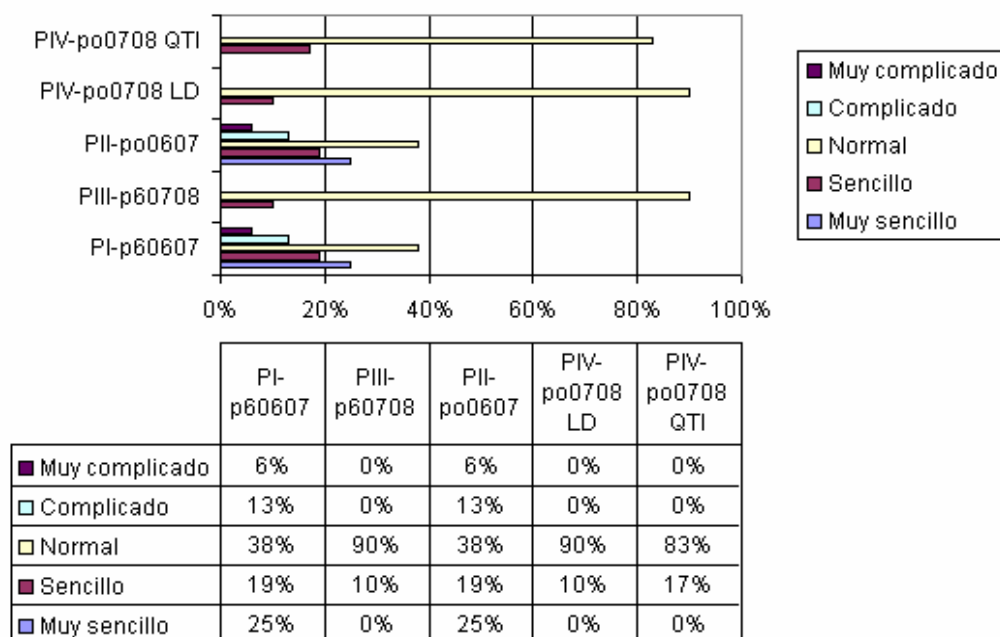


Figura 86: Resultados asociados al proceso de desarrollo con las herramientas

Tiempo empleado en el desarrollo

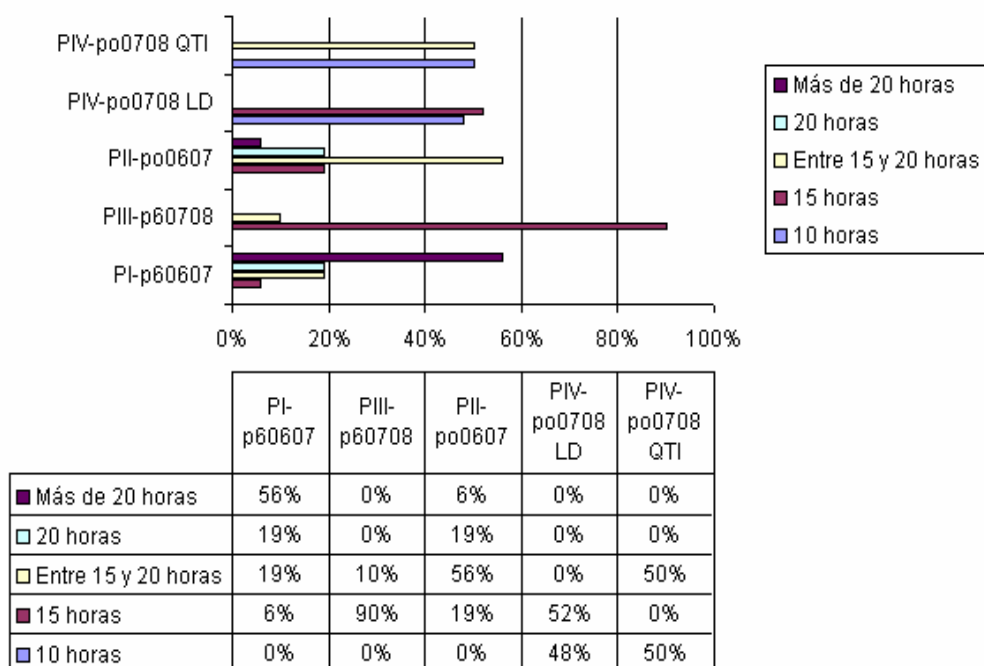


Figura 87: Resultados del estudio del tiempo empleado en el desarrollo

9.2. Conclusiones del Proyecto

La realización de este Proyecto de Fin de Carrera ha permitido el estudio de un campo desconocido por el autor hasta la fecha, el E-Learning, una disciplina que aun que se encuentra en un punto muy temprano de desarrollo, tiene muchas posibilidades de ampliación, puesto que con el incipiente uso de las tecnologías, que con el tiempo se van introduciendo cada vez más en las actividades rutinarias de la gente, haciendo desaparecer incluso algunas actividades que tradicionalmente se realizaban por otros medios, va a permitir que el proceso de enseñanza sea mucho más dinámico, puesto que la posibilidad de crear unidades de aprendizaje que pueden ser consultadas y utilizadas por los alumnos desde su casa va a dar la posibilidad de que su formación sea mucho más intensa y profunda que hasta ahora, que estaba limitada a la presencia física en un lugar determinado de tutor y alumno una limitación muy importante aunque a primera vista parezca una minucia en este asunto.

Ya desde un punto de vista tecnológico, el Proyecto me ha permitido adquirir unos conocimientos más profundos de aplicaciones Web, ya que el uso de muchos frameworks que no había visto en mi formación académica ha hecho que le pierda el “miedo” a la utilización de cosas nuevas, y además he podido poner en práctica muchos de los conocimientos teóricos que en la carrera no había podido poner en práctica nada más que en pequeños aplicativos, como los patrones de diseño o la aplicación del Modelo-Vista-Controlador, un tipo de desarrollo que se usa mucho en el mundo laboral.

Por lo tanto, el Proyecto me ha sido de gran ayuda para completar mi formación, permitiendo además que desarrolle el interés por el campo del E-Learning, en el que si se implanta un estándar para todos, se pueden plantear múltiples sistemas como los que se van a enumerar en la sección de “*Ampliaciones Futuras*”, en las que sin duda estaría interesado en seguir su desarrollo. Además, el otro gran problema que me he encontrado es la relativamente pequeña cantidad de documentación que hay sobre el tema, sobre todo en castellano, además de que el framework CopperCore se encuentra un poco parado en su desarrollo, lo que es comprensible por el estancamiento que sufre el proceso de estandarización.

En mi opinión, el campo del E-Learning debe ser potenciado buscando su implantación definitiva, puesto que sus ventajas son inmensas y se producirá una importante mejora con respecto a la situación que se vive en el mundo de la enseñanza.

9.3. Ampliaciones Futuras

Este proyecto fin de carrera deja abiertas algunas líneas para trabajo futuro y admite varias ampliaciones. Podemos citar algunas líneas de trabajo.

- Mejorar la edición de los contenidos que facilite la creación desde cero de contenidos educativos así como su posible modificación, de una manera sencilla y atractiva para el usuario.

- En relación a las búsquedas de los contenidos sería deseable ampliar el abanico a otros repositorios accesibles vía servicios Web, tal como se indica en la solución **MD2: Método de desarrollo de materiales didácticos**, de esta manera la búsqueda de contenidos no se limitaría a un repositorio local y la localización de contenidos sería más enriquecedora.

- En cuanto a la descripción de los contenidos, se podría ampliar la restricción inicial de centrarnos en una sola área de conocimiento “Ciencias de la Computación”, descrita según los planes de estudio de la ACM (Association for Computing Machinery). Sino que teniendo en consideración las recomendaciones de posibles ampliaciones de la solución MD2 se extienda a otras áreas de conocimiento descritas según ACM y el sistema de catalogación de la Librería del Congreso de los Estados Unidos (LLC).

- En relación a la accesibilidad de la aplicación se podrían aprovechar las características de internacionalización del Framework Struts, para que la herramienta no se centre en un solo idioma. Además de facilitar diferentes hojas de estilo CSS que permitan la visualización de la información de la aplicación para usuarios con problemas de visión.

10. Bibliografía

Este apartado incluye la bibliografía, escrita y contenido WEB, que se ha utilizado principalmente como apoyo en la realización de este Proyecto de Fin de Carrera. Se han consultado muchas más fuentes, pero se incluyen las más relevantes y las que se mencionan en el documento desarrollado:

- [1] - Estado actual de los sistemas e-learning. Francisco José García Peñalvo, http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm.
- [2] - American Society of Training and Development, <http://www.astd.org/lc>
- [3] - ROSENBERG, M. J. (2001) E-learning strategies for delivering knowledge in the digital age. McGraw-Hill. Marc Rosenberg, <http://www.marcrosenberg.com/>
- [4] - LOZANO GALERA, J. (2004) El triángulo del e-learning. <http://www.noticias.com>.
- [5] - APPEL, M. C.; BLIN, F. (2003). "Focus on Form and ICT in Language Learning and Teaching".
- [6] - IMS Global Learning Consortium. Learning Design Specification. <http://www.imsglobal.org/learningdesign/>
- [7] - Gestipolis. <http://www.gestipolis.com/administracion-estrategia/estrategia/que-es-el-e-learning.htm>
- [8] - Evaluation and Selection of Web Course Management Tools, <http://www.sunilhazari.com/education/webct/>
- [9] - Blackboard, <http://www.blackboard.com/>
- [10] - Moodle.org: open-source community-based tools for learning <http://moodle.org/>
- [11] - .LRN Home, <http://dotlrn.org/>
- [12] - Estándares eLearning, <http://internet-educativa.pbworks.com/Est%C3%A1ndares-eLearning>
- [13] - MD2 Method: The Didactic Materials Creation from a Model Based Perspective, 2006, Carmen L. Padrón, Paloma Díaz y Ignacio Aedo

[14] - World Wide Web Consortium, <http://www.w3c.es/>

[15] – GML, Geography Markup Language,
<http://www.opengeospatial.org/standards/gml>

[16] – XML Core Working Group, <http://www.w3.org/XML/Core/>

[17] – JDOM, <http://www.jdom.org/>

[18] – SAX Project, <http://www.saxproject.org/>

[19] – Jboss Community, <http://www.jboss.org/>

[20] – Apache Struts, <http://struts.apache.org/>

[21] – CopperCore, <http://coppercore.sourceforge.net/>

[22] - SLeD Learning Design Placer, <http://sled.open.ac.uk/>

[23] – Model-View-Controller,
<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

[24] – European Space Agency, <http://www.esa.int/esaCP/index.html>

[25] – Ministerio de Industria, Turismo y Comercio, <http://www.boe.es/>

[26] - Padrón-Nápoles, C. L. (2009) “Desarrollo de materiales didácticos desde una perspectiva basada en modelos” Tesis Doctoral. Programa Doctorado Ciencia y Tecnología Informática. Universidad Carlos III de Madrid

[27] – Iván Peláez (2008) “MD2: Sistema de búsqueda y composición de materiales basado en IMS QTI”. Proyecto Fin de Carrera dirigido por Carmen L. Padrón Nápoles

Otra bibliografía consultada:

- E-learning teleformación: diseño, desarrollo y evaluación de la formación a través de Internet, Carlos Marcelo García, Edicions Gestió 2000, S.A.
- E-Learning : implantación de proyectos de información On-Line, Eva I. Fernández Gómez, Ra-Ma, Librería y Editorial Microinformática
- Teoría y práctica del E-learning, Marcela Paz González Brignardello, Grupo Eidos.

- E-Learning: Strategies for Delivering Knowledge in the Digital Age, Marc J. Rosenberg, McGraw-Hill Trade

Otras referencias WEB consultadas

- Ministerio de Educación, <http://www.educacion.es/portada.html>
- Comunidad ELearning, <http://www.elearningworkshops.com/>
- Portal Andaluz de e-Learning - Junta de Andalucía,
<http://prometeo3.us.es/publico/index.jsp>
- E-Learning Centre, <http://www.e-learningcentre.co.uk/>
- Alex Little BLOG, <http://alexlittle.net/blog/tag/sled/>
- Juliette White BLOG, <http://www.open.ac.uk/blogs/d4ld/?p=3>